

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



**FEUP**

# **Estudo e aplicação da tecnologia Oracle BPEL - Pedidos de Autorização de Despesa - PAD**

**João Filipe Machado Marques**

Relatório de Projecto/Dissertação

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Prof. Gabriel David (Professor Associado)

13 de Julho de 2009

# **Estudo e aplicação da tecnologia Oracle BPEL - Pedidos de Autorização de Despesa - PAD**

**João Filipe Machado Marques**

Relatório de Projecto/Dissertação

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: José Manuel Magalhães Cruz (Professor Auxiliar)

---

Arguente: José Manuel Moreira (Professor Auxiliar Convidado)

Vogal: Prof. Gabriel David (Professor Associado)

13 de Julho de 2009

# Resumo

A evolução das tecnologias, aumento da diversidade de aplicações e constantes mudanças do mercado tornou fulcral que os sistemas de informação suportem rápidas e eficazes alterações. Este requisito foi uma das motivações da arquitectura orientada a serviços na construção de aplicações assentes na web.

A realização deste projecto inclui reformulação parcial do módulo de suporte do Oracle BPEL Process Manager e a constituição de um processo de negócio tendo por base os conceitos e objectivos da arquitectura orientada a serviços. Desta forma, foi necessário analisar e perceber o funcionamento de uma nova linguagem de definição e automatização de processos de negócio, denominada Web Services Business Process Execution Language WS-BPEL e a respectiva implementação Oracle. O módulo a reformular faz parte de um sistema de informação académico (SIGARRA). Parte da estrutura necessária já se encontrava implementada, sendo necessário um estudo prévio da mesma.

Este relatório documenta os esforços da criação da estrutura de interligação do Oracle BPEL Process Manager ao SIGARRA e a conversão do módulo de Pedidos de Autorização de Despesa em Oracle Workflow para BPEL. Este processo inclui todos os passos necessários que uma universidade efectua para realizar uma despesa.

A conversão foi realizada em duas fases. Numa primeira fase reviu-se e completou-se a estrutura de interligação referida. Na segunda fase refez-se o desenho dos processos de PADs em BPEL, recorrendo aos mecanismos de interligação.

O resultado obtido permite concluir que a substituição do Oracle Workflow por Oracle tornou mais distribuída a interface com o SIGARRA o que resulta numa maior flexibilidade e integração com o motor do BPEL e o sistema de informação.

# Abstract

Information systems need so support business changes quickly and efficiently. However they also need to adapt to the fast development of new technologies. To manage problems related to changing requirements, technology development and integration, different methods have been proposed. Service-oriented architecture is the latest architectural approach related to integration development and maintenance of complex enterprise information systems.

This project includes the review and update of the Oracle BPEL Process Manager integration architecture and the recasting a business process by implementing the concepts and objectives of the SOA Architecture. For this, it was necessary to analyze and find solutions to integrate a new language for definition and automation of business processes, called Web Services Business Process Execution Language WS-BPEL and study the Oracle implementation. The module needed to be reviewed and updated was part of the information system (SIGARRA).

This report documents the efforts needed to create the interconnection architecture between Oracle BPEL Process Manager and the informations systems and the recasting of a business process, cost request. This process includes all necessary steps that a university in Portugal needs to make a cost.

The project was done in two different moments. On the first, all the underlying infrastructure that allows integration between BPEL and the information system was reviewed and updated to support new features. After that, recasting a business process *cost request* using the integration framework previously mentioned.

This project concludes that reformulating the entire engine and interfaces to integrate them with the new distributed architecture results in greater flexibility and integration between Oracle BPEL Process Manager and information system.

# Agradecimentos

Em primeiro lugar, agradeço a toda a equipa do PSI em especial ao Engenheiro Luís Silva, Engenheiro Gil Silva, Engenheiro Filipe Silva e ao Engenheiro António Cunha de quem obtive um total apoio e auxílio fundamentais para a elaboração deste trabalho.

Um agradecimento especial ao Engenheiro Manuel Machado, pela confiança depositada e pelas oportunidades fornecidas.

Gostaria também de agradecer ao Professor Doutor Gabriel David, pela sua disponibilidade para a orientação do meu projecto de estágio.

Agradeço os meus colegas Hugo Zenha, Vera Francisco e Filipe Silva pelo apoio, companhia e bons momentos passados.

Por fim, um especial agradecimento a todos os meus amigos e minha família, especialmente aos meus pais, João Marques e Rosa Marques, e ao meu irmão Diogo Marques, pelo apoio e suporte que me permitiram ultrapassar diversas dificuldades e incutir o estudo, o trabalho e a realização profissional como os passos fundamentais para um futuro percurso cheio de sucessos.

João Filipe Machado Marques

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	1
1.2	A Instituição . . . . .	2
1.3	Projecto . . . . .	2
1.4	Motivação e Objectivos . . . . .	3
1.5	Estrutura do documento . . . . .	3
1.6	Planeamento . . . . .	3
<b>2</b>	<b>Arquitectura Orientada a Serviços</b>	<b>6</b>
2.1	BPEL . . . . .	8
2.1.1	Funcionalidades . . . . .	10
2.1.2	Exemplo de um Processo de Negócio . . . . .	11
2.1.3	Aplicações e servidores disponíveis . . . . .	14
2.2	Oracle BPEL Process Manager . . . . .	15
2.2.1	Arquitectura do Oracle BPEL Process Manager . . . . .	16
2.2.1.1	Desenho de Processos . . . . .	16
2.2.1.2	Servidor BPEL . . . . .	17
2.2.1.3	Consola BPEL . . . . .	19
2.2.1.4	Base de Dados . . . . .	20
2.2.2	Conclusão . . . . .	21
<b>3</b>	<b>Especificação do módulo</b>	<b>22</b>
3.1	Descrição . . . . .	22
3.1.1	Edição . . . . .	23
3.1.2	Validações . . . . .	23
3.1.2.1	Validação do responsável da despesa . . . . .	23
3.1.2.2	Validação do responsável do centro de custo . . . . .	23
3.1.3	Cabimento . . . . .	23
3.1.4	Validação administrativa . . . . .	23
3.1.5	Compromisso . . . . .	24
3.1.6	Inserção de facturas . . . . .	24
3.1.7	Registo de terceiros . . . . .	24
3.1.8	Pagamento . . . . .	24
3.2	Requisitos . . . . .	24
3.2.1	Funcionais . . . . .	24
3.2.2	Não funcionais . . . . .	25
3.2.3	Casos de utilização . . . . .	26

## CONTEÚDO

3.2.3.1	Actores . . . . .	26
3.2.3.2	Pacotes . . . . .	26
<b>4</b>	<b>Implementação</b>	<b>38</b>
4.1	Arquitectura Lógica do Sistema . . . . .	38
4.1.1	Decomposição Horizontal . . . . .	38
4.1.1.1	Interface . . . . .	38
4.1.1.2	Lógica de Negócio . . . . .	39
4.1.1.3	Acesso a Dados . . . . .	40
4.1.2	Decomposição Vertical . . . . .	40
4.2	Arquitectura Física do Sistema . . . . .	41
4.3	Sistema SIBPEL . . . . .	42
4.3.1	Serviço de identidade . . . . .	42
4.3.2	Sensores . . . . .	43
4.3.3	Workflow . . . . .	44
4.4	Novo módulo - Pedidos de Autorização de Despesa . . . . .	44
4.4.1	Definição dos Processos de Negócio BPEL . . . . .	44
4.4.1.1	Editar PAD . . . . .	45
4.4.1.2	Validação responsável da despesa . . . . .	46
4.4.1.3	Validação do Centro de Custo . . . . .	46
4.4.1.4	Cabimento . . . . .	46
4.4.1.5	Validação Administrativa . . . . .	47
4.4.1.6	Compromisso . . . . .	47
4.4.1.7	Inserção de facturas . . . . .	47
4.4.1.8	Registo de terceiros . . . . .	50
4.4.1.9	Pagamento . . . . .	50
4.4.1.10	PAD . . . . .	51
4.4.2	Modelo relacional de Dados . . . . .	53
4.4.3	Motor de lógica . . . . .	53
4.4.4	Interface . . . . .	53
<b>5</b>	<b>Resultados</b>	<b>57</b>
5.1	Avaliação Global do Projecto . . . . .	57
5.2	Caso de Utilização . . . . .	58
5.3	Comparação entre BPEL e Workflow . . . . .	64
<b>6</b>	<b>Conclusões e Perspectivas de Trabalho Futuro</b>	<b>65</b>
6.1	Satisfação dos Objectivos . . . . .	65
6.2	Trabalho Futuro . . . . .	66
	<b>Referências</b>	<b>67</b>
<b>A</b>	<b>Diagramas</b>	<b>68</b>
A.0.1	Diagrama de fluxo . . . . .	68
A.0.2	Modelo de Dados . . . . .	68

# Lista de Figuras

1.1	Planeamento . . . . .	5
2.1	Visão geral da Arquitectura SOA . . . . .	8
2.2	Tecnologias do Projecto integradas na Arquitectura SOA . . . . .	8
2.3	Exemplo da agregação de serviços num processo de negócio . . . . .	9
2.4	Arquitectura do Oracle BPEL Process Manager . . . . .	17
2.5	Oracle JDeveloper . . . . .	18
2.6	Oracle BPEL Console . . . . .	20
2.7	Oracle Worklist . . . . .	20
3.1	Processo de PAD . . . . .	28
3.2	Fase: Edição . . . . .	29
3.3	Validações . . . . .	30
3.4	Fase: Validação do responsável da despesa . . . . .	30
3.5	Fase: Validação do responsável do Centro de Custo . . . . .	31
3.6	Fase: Cabimento . . . . .	31
3.7	Fase: Validação Administrativa . . . . .	32
3.8	Fase: Compromisso . . . . .	32
3.9	Fase: Inserção de Facturas . . . . .	33
3.10	Fase: Registo de Terceiros . . . . .	34
3.11	Fase: Pagamento . . . . .	35
3.12	Casos de utilização gerais . . . . .	36
3.13	Casos de utilização por pacotes . . . . .	37
4.1	Decomposição Horizontal . . . . .	39
4.2	Decomposição Vertical . . . . .	41
4.3	Arquitectura Física . . . . .	42
4.4	Arquitectura SIBPEL . . . . .	43
4.5	Edição do Pad . . . . .	45
4.6	Validação responsável da despesa . . . . .	46
4.7	Validação do Centro de Custo . . . . .	47
4.8	Cabimento . . . . .	48
4.9	Validação Administrava . . . . .	48
4.10	Compromisso . . . . .	49
4.11	Inserção de facturas . . . . .	49
4.12	Registo de terceiros . . . . .	50
4.13	Validação do Centro de Custo . . . . .	51
4.14	Pedido de Autorização de Despesa . . . . .	52



## LISTA DE FIGURAS

4.15	Criação do pedido de autorização de despesa fase 1 . . . . .	54
4.16	Criação do pedido de autorização de despesa fase 2 . . . . .	54
4.17	Visualização dos detalhes de pedido de autorização de despesa . . . . .	55
4.18	Edição de detalhes contabilísticos . . . . .	55
4.19	Visualização dos detalhes contabilísticos . . . . .	56
4.20	Inserção de facturas . . . . .	56
4.21	Exemplo de notificação no gestor de tarefas . . . . .	56
5.1	Criação do pedido de autorização de despesa parte 1 . . . . .	58
5.2	Criação do pedido de autorização de despesa parte 2 . . . . .	58
5.3	Detalhes do pedido de autorização de despesa . . . . .	59
5.4	Detalhes do pedido de autorização de despesa . . . . .	59
5.5	Notificação de validação do responsável da despesa . . . . .	60
5.6	Notificação de validação do responsável da centro de custo . . . . .	60
5.7	Notificação de contabilidade informando o que é necessário efectuar o processo de cabimentação . . . . .	60
5.8	Cabimentação da despesa . . . . .	61
5.9	Notificação de validação do conselho administrativo . . . . .	61
5.10	Notificação de autor para efectuar a compra . . . . .	62
5.11	Inserção online de facturas . . . . .	62
5.12	Notificação de registo de terceiros . . . . .	63
5.13	Notificação de registo de terceiros . . . . .	63
5.14	Integração da ordem de pagamento . . . . .	63
5.15	Autorização do pagamento . . . . .	64
A.1	Diagrama de fluxo do processo de Pedidos de Autorização de despesa . . .	69
A.2	Modelo de dados do módulo dos Pedidos de Autorização de Despesa . . .	70
A.3	Modelo de dados da integração do bpel com o módulo dos Pedidos de Autorização de Despesa . . . . .	71

# Lista de Tabelas

4.1	Lista de operações implementadas no serviço de identidade . . . . .	43
4.2	Lista de operações implementadas no serviço de identidade . . . . .	44

# Abreviaturas e Símbolos

MIEIC	Mestrado Integrado em Engenharia Informática
FEUP	Faculdade de Engenharia da Universidade do Porto
UP	Universidade do Porto
CICA	Centro de Informática Prof. Correia de Araújo
PSI	Projecto de Sistemas de Informação
SIGARRA	Sistema de Informação para Gestão Agregada dos Recursos e Registos Académicos
SI	Sistema de Informação
PAD	Pedido de Autorização de Despesa
SOA	Service-oriented architecture
XML	Extensible Markup Language
BPEL	Business Process Execution Language
WS-BPEL	Web Services Business Process Execution Language
WSDL	Web Services Description Language
SOAP	Simple Object Access Protocol
WSFL	Web Services Flow Language
IBM	International Business Machines
OASIS	Organization for the Advancement of Structured Information Standards
JEE	Java Enterprise Edition
EJB	Enterprise Java Beans
RMI	Remote Method Invocation
JMS	Java Message Service
WSIF	Web Services Invocation Framework
PL/SQL	Procedural Language/Structured Query Language
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
UML	Unified Modeling Language
XPATH	XML Path Language
API	Application Programming Interface
UDDI	Universal Description Discovery and Integration
JDBC	Java Database Connectivity
OC4J	Oracle Containers for J2EE
ESB	Enterprise Service Bus

# Capítulo 1

## Introdução

O presente documento apresenta os detalhes da realização do projecto “Estudo e aplicação da tecnologia Oracle BPEL - Pedidos de Autorização de Despesa - PAD”. Foi desenvolvido no contexto do Projecto de Sistemas de Informação (PSI) da Faculdade de Engenharia da Universidade do Porto e elaborado por João Filipe Machado Marques, no âmbito do projecto final do Mestrado Integrado em Engenharia Informática e Computação (MIEIC) para a Faculdade de Engenharia da Universidade do Porto.

No decurso deste documento é exposto o propósito e motivação do projecto, o seu impacto, passando pelo planeamento, análise, desenho, desenvolvimento e testes efectuados e terminado nos resultados obtidos.

Este primeiro capítulo constitui uma introdução ao projecto e uma apresentação da instituição onde foi realizado. Inclui ainda a estrutura de capítulos que compõem este documento.

### 1.1 Enquadramento

O desenvolvimento do projecto “Estudo e aplicação da tecnologia Oracle BPEL - Pedidos de Autorização de Despesa - PAD” foi proposto para a realização de um projecto final, conforme o estipulado pelo actual plano de estudos para os alunos finalistas do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

A sua realização decorreu durante o segundo semestre do ano lectivo 2008/2009, com início em Março de 2009 e fim em Junho de 2009, no Projecto de Sistemas de Informação (PSI) na Faculdade de Engenharia da Universidade do Porto.

O autor deste projecto é o aluno João Marques, finalista do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do

Porto, contando ainda com a orientação e supervisionamento do Professor Gabriel Torcato David, por parte da FEUP, e do Engenheiro Manuel Machado por parte do PSI.

### 1.2 A Instituição

O presente projecto de estágio foi proposto pela Faculdade de Engenharia da Universidade do Porto e desenvolvido no Projecto de Sistemas de Informação (PSI) nesta faculdade. É constituída maioritariamente por colaboradores da FEUP pertencentes aos quadros do Centro de Informática Professor Correia de Araújo (CICA), cujas funções e responsabilidades distribuem-se pelas diversas áreas de interesse desta instituição.

O seu principal objectivo é o desenvolvimento do Sistema de Informação para Gestão Agregada dos Recursos e Registos Académicos (SIGARRA). Consiste num sistema de informação integrado que permite facilitar o acesso à informação relevante de uma instituição, de carácter pedagógico, científico, técnico ou administrativo, bem como dinamizar a colaboração interna e com as comunidades académicas e empresariais exteriores. Devido à forte aposta na integração das mais recentes tecnologias, rapidamente se destacou como um sistema fulcral, actualmente utilizado por diversas instituições de ensino em Portugal.

Tendo como principais objectivos, o desenvolvimento de produtos e serviços na área de Sistemas de Informação, dos seus sistemas destacam-se, para além do SIGARRA, a Gestão Integrada de Contra-Ordenações para a Câmara Municipal do Porto e o Sistema de Informação do Sindicato de Trabalhadores da Função Pública do Norte.

### 1.3 Projecto

Este projecto tem como título “Estudo e aplicação da tecnologia Oracle BPEL - Pedidos de Autorização de Despesa - PAD”. Consiste na passagem de um processo de negócio, pedidos de autorização de despesa (PAD), implementado em Oracle Workflow para Oracle BPEL. Este processo é responsável por enviar pelas diversas entidades solicitações para efectuar despesas, registar eventos contabilísticos e efectuar os pagamentos.

Actualmente, os processos de negócio encontram-se desenvolvidos em Oracle Workflow. Oracle Workflow é uma aplicação constituída por vários componentes, que tem como objectivo auxiliar e otimizar a gestão de processos de negócio de uma instituição. É desenvolvido em PL/SQL e neste momento é um produto descontinuado. Desta forma, é necessário a médio prazo, ser substituído por outra tecnologia.

Para esta conversão a tecnologia seleccionada foi o padrão baseado em Web Services, denominado por Business Process Execution Language (BPEL). BPEL adopta o WS-BPEL 2.0 OASIS Standard - uma nova linguagem para desenho e informatização de processos de negócio apoiada pela arquitectura orientada a serviços (SOA). Incorpora as mais

recentes tecnologias tais como a Web Services Description Language (WSDL), XML e comunicações SOAP. Para a implementação, a ferramenta seleccionada foi o BPEL Process Manager. Permite uma gestão e desenho de processos de negócio assente em BPEL.

Aquando do início do projecto já se encontrava criada uma arquitectura de integração e interligação entre o SIGARRA e o Oracle BPEL Process Manager, o SIBPEL (desenvolvida por Ivo Marinho [dRM08]). Foi realizado um estudo sobre as características e funcionalidades desta estrutura sendo necessário proceder a alterações de forma a tornar mais completa e incluir os requisitos do processo.

### 1.4 Motivação e Objectivos

O desenvolvimento deste projecto decorreu segundo os seguintes objectivos:

- **Levantamento de requisitos do módulo de Pedidos de Autorização de Despesa** - pretende-se efectuar o levantamento das actuais funcionalidades deste módulo e, de novos casos de utilização junto dos seus principais utilizadores.
- **Completar a camada de integração** - é pretendido rever e completar toda a estrutura de interligação entre o BPEL e o sistema de informação (SIGARRA).
- **Transformação do processo de Pedidos de Autorização de Despesa em Oracle Workflow para BPEL** - é requerido que o processo de PAD em Workflow seja convertido para BPEL. Deve tirar-se o máximo partido das vantagens do BPEL dando uso à arquitectura implementada.

### 1.5 Estrutura do documento

Este documento é composto por seis capítulos. No capítulo 2 são apresentados os conceitos e vantagem da arquitectura orientada a serviços e detalhando uma das suas realizações a tecnologia BPEL. O capítulo 3 é descreve detalhadamente o problema e o seu contexto. O capítulo 4 é descrita a implementação do projecto bem como todas as decisões. No capítulo 5 é apresentado os resultados do projecto e evidenciado o estado da implementação proposta. Por fim, no capítulo 6 faz-se um balanço ao que falta elaborar e demonstrando possibilidades de evolução.

### 1.6 Planeamento

O desenvolvimento deste projecto foi dividido em quatro fases: especificação de requisitos, estudo da tecnologia, implementação e relatório final.

## Introdução

- **25 de Fevereiro a 6 de Março** - Especificação de requisitos novos e implementados.
- **5 de Março a 18 de Março** - Estudo da tecnologia (tutoriais, exemplos e estudo do backend do Oracle BPEL Process Manager).
- **9 de Março a 5 de Maio** - Implementação (correção de erros do modulo actual de PADs, estudo e elaboração dos requisitos do backend do módulo de PADs, implementação do backend do modulo de PADs, implementação do módulo DataPublisherService e interfaces).
- **1 de Junho a 29 de Junho** - Relatório final.

Na figura [1.1](#) é apresentada o diagrama com a calendarização.

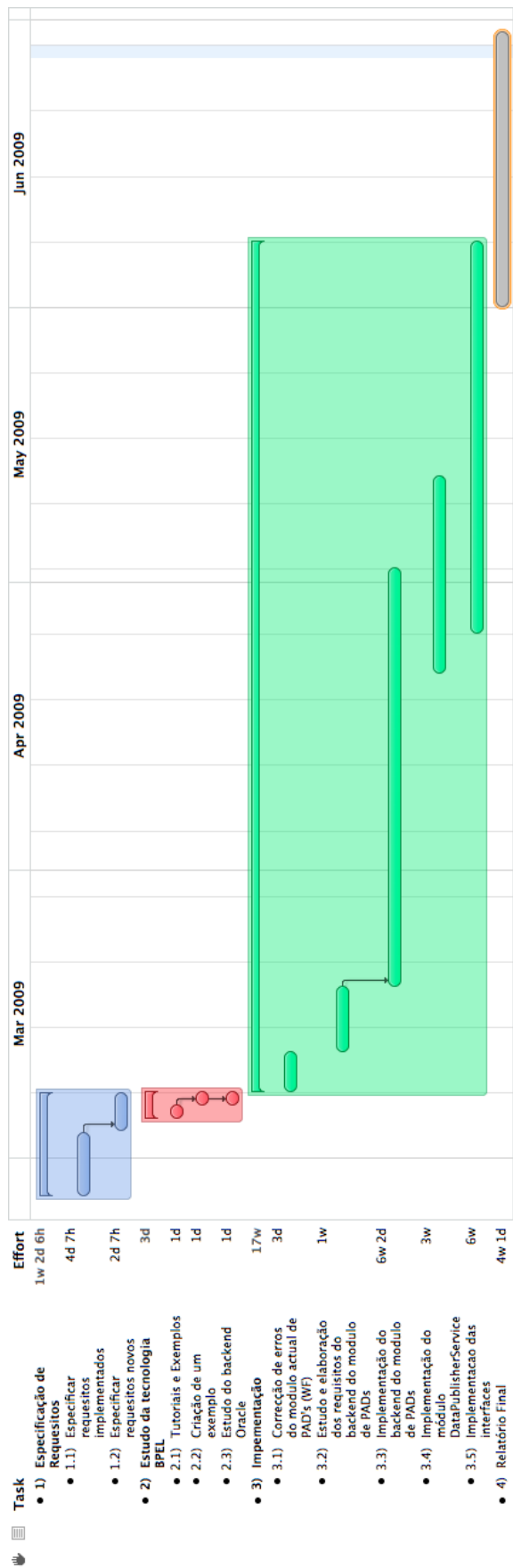


Figura 1.1: Planeamento



## Capítulo 2

# Arquitectura Orientada a Serviços

Com a evolução das tecnologias e aumento da diversidade de aplicações, tornou-se crucial que os sistemas de informação suportem alterações rápidas e eficazes, de modo a adaptarem-se às mudanças dos mercados. Estas alterações não são só efectuadas ao nível dos processos de negócio, mas também como adaptações às mais recentes tecnologias.

A maioria das empresas contém uma grande diversidade de sistemas de informação, aplicações, tecnologias e arquitecturas. Garantir a integração deste conjunto de sistemas heterogéneos num único sistema centralizado, torna-se fulcral no acesso e controlo de dados e verificações de integridade.

Para resolver estas metas, a Arquitectura Orientada a Serviços (SOA) tornou-se na resposta mais eficaz existente actualmente a nível de integração, desenvolvimento e manutenção de complexos sistemas de informação. É uma abordagem que ajuda a manter os sistemas escaláveis e flexíveis, enquanto estes aumentam e crescem de complexidade.

A arquitectura SOA fornece métodos para desenvolvimento de sistemas de informação e integração, onde cada funcionalidade é exposta como um serviço. Permite a reutilização de aplicações e sistemas existentes, interoperabilidade e integração de aplicações e, composição de processos de negócio a partir de serviços (funcionalidades) fornecidos pelas aplicações. Um dos objectivos principais desta arquitectura é garantir que futuras alterações e manutenção de sistemas de informação sejam efectuadas de forma fácil e rápida.

Uma forma de disponibilização destes serviços são os web services. Correspondem a um conjunto de definições que operam sobre a troca de mensagens entre clientes e servidores, contendo dados definidos em XML, seguindo os padrões de comunicação SOAP. A interface de comunicação é definida pelo protocolo Web Services Description Language (WSDL). [Jos07]

Para suportar estas funcionalidades, esta arquitectura define todo um conjunto de conceitos, arquitecturas e frameworks, reduzindo a complexidade e apostando na integração

e reutilização, de modo, a garantir um menor custo de desenvolvimento, integração e de manutenção possível. Estes conceitos não se apoiam apenas numa única tecnologia, mas sim num conjunto que poderá ser agrupado do seguinte modo:

- **Serviços** - fornecem funcionalidades do negócio de forma autónoma e transparente.
- **Interfaces** - define o conjunto de operações públicas disponibilizadas pelo serviço prestador aos consumidores.
- **Mensagens** - operações definidas como um conjunto de mensagens. Especificam os dados que são trocados independentemente da plataforma ou linguagem.
- **Sincronização** - os consumidores acedem aos serviços através de um protocolo como o SOAP ou Enterprise Service Bus (ESB) que serve como camada de comunicação. Estas comunicações podem ser síncronas ou assíncronas.
- **Minimização de Dependências** - expõe só as dependências necessárias, garantindo que futuras actualizações apenas necessitam de um pequeno número de alterações.
- **Registos** - para simplificar e automatizar o processo de pesquisa, os serviços são mantidos em registos. A pesquisa é feita através do nome, funcionalidade ou propriedades. O Universal Description, Discovery and Integration (UDDI) é um exemplo de um registo de serviços.
- **Qualidade dos Serviços** - atributos necessários que os serviços devem possuir, tais como: segurança, fiabilidade e gestão das transacções, correlação e especificação.
- **Composição de Serviços em Processos de Negócio** - o conceito de SOA corresponde à composição de serviços em processos de negócio. Serviços são compostos e orquestrados numa ordem específica e seguem um conjunto de regras que permitem criar processos de negócio. Esta composição permite aceder de forma fácil e rápida aos processos de negócio, para além de permitir futuras alterações e suporte para sistemas com requisitos em constante mudança. É nesta fase que é introduzida uma linguagem dedicada, Business Process Execution Language (BPEL), suportada por um servidor onde as definições dos processos de negócio são executadas.

A figura 2.1 apresenta uma visão geral da arquitectura SOA e dos conceitos anteriormente referidos:

Se na figura 2.1 substituírmos os conceitos pelas principais tecnologias e estabelecermos a respectiva correspondência, consegue-se obter uma visão clara de como o BPEL se relaciona e integra na arquitectura SOA (figura 2.3).

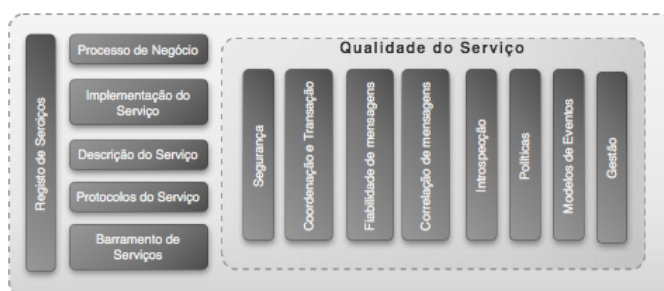


Figura 2.1: Visão geral da Arquitetura SOA



Figura 2.2: Tecnologias do Projecto integradas na Arquitetura SOA

## 2.1 BPEL

Desenvolvido sobre uma infra-estrutura baseada em standards abertos e mantendo-se assente nos princípios de SOA, o Business Process Execution Language (BPEL) é a última inovação para a composição e orquestração de serviços. Fornece a habilidade de exprimir processos de negócio de forma standard, fácil e flexível. Permite igualmente que os processos sejam automatizados suportando de forma rápida alterações necessárias, diminuindo assim, os custos e complexidade.

A tecnologia BPEL foi introduzida para completar a realização dos conceitos da arquitectura SOA. Esta especifica que os serviços devem ser agregados em novos e maiores serviços que, depois, serão integrados em processos de negócio. Desta forma, poder-se-á definir um processo de negócio, como um conjunto de actividades através dos quais os serviços são invocados. No entanto, o próprio processo de negócio deverá ser visto para o exterior igualmente como um novo serviço e igual a todos os outros, que são por ele invocados. Uma definição generalizada para os processos de negócio poderá ser:

*Um processo de negócio é um conjunto de invocações coordenadas de serviços e de actividades relacionadas, que produzem um resultado final, quer numa simples organização ou em várias. [Jur06]*

Para ilustrar a composição de um possível processo de negócio, retirado de [Jur06], apresenta-se o exemplo onde um cliente efectua um pedido a uma agência de viagens, para obter o melhor preço para uma dada viagem. A agência, por sua vez, consulta os serviços disponibilizados por várias companhias aéreas em forma de Web Services, sobre as suas disponibilidades e preços. Por fim, calcula o melhor preço e devolve a proposta ao cliente. A composição de actividades e serviços deste possível processo de negócio é apresentada na figura 2.3.

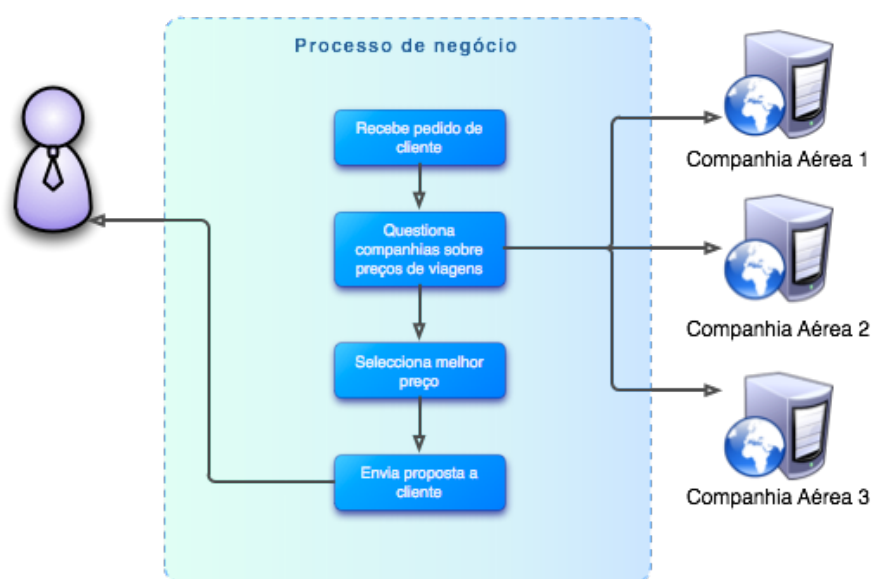


Figura 2.3: Exemplo da agregação de serviços num processo de negócio

Este processo de negócio constitui um simples exemplo real da integração da tecnologia BPEL na definição de processo de negócio. Agrupa e reutiliza serviços externos à empresa, de uma forma fácil e transparente do ponto de vista do processo de negócio. Ou seja, o consumidor não tem acesso à forma como os serviços internos ao processo retornam os valores, quer seja, por acessos ao sistema, a ficheiros, à base de dados, ou a qualquer outra forma. O que interessa é apenas os valores que o processo devolve.

Os processos de negócio em BPEL são também divulgados e acedidos como serviços. Tiram o máximo partido dos conceitos e objectivos do SOA na reutilização e diminuição da complexidade dos sistemas. Foi desta forma que a especialização da linguagem BPEL se tornou, actualmente, no standard dominante da arquitectura SOA para a automação de processos de negócio utilizando Web Services.

*O BPEL define processos de negócio como agrupamentos de serviços, operações e*

*outros processos e expõe as suas funcionalidades como serviços.* [Jur06]

Entre organizações, permite uma mais fácil e eficaz integração com parceiros comerciais. Estimula a organização a definir os processos de negócio, que por sua vez levam à optimização, reengenharia e selecção.

A primeira versão BPEL4WS 1.0 foi desenvolvida em conjunto pela IBM, BEA e a Microsoft em Agosto de 2002. Baseou-se em duas principais linguagens de workflow e definição de processos: o WSFL (Web Services Flow Language) da IBM e XLANG da Microsoft. Mais tarde, SAP e Siebel juntaram-se ao grupo citado anteriormente, da qual resultaram diversas modificações e melhoramentos, levando à adopção da versão 1.1 em Março de 2003. Posteriormente foi submetido como standard para a OASIS (Organization for the Advancement of Structured Information Standards) onde foi formado o WS-BPEL, actualmente na versão 2.0. [Jos07]

### **2.1.1 Funcionalidades**

A linguagem BPEL usa um vocabulário XML para especificar e descrever processos de negócio desde os mais simples até aos mais complexos. Em certa medida, BPEL é similar a linguagens de programação tradicionais. Oferece variáveis, estruturas, ciclos, atribuições, verificações, etc que permitem definir processos de forma algorítmica. BPEL é uma linguagem especializada na definição de processos de negócio. Por outro lado, é menos complexa que as linguagens de programação tradicionais, o que simplifica a aprendizagem.

O aspectos fundamentais desta tecnologia são a simplicidade de invocar Web Services tanto síncronos como assíncronos, bem como invocar operações sequenciais ou paralelas. Além disso, proporciona um vocabulário rico em gestão de erros e excepções, actividades de espera, suporte para a execução de longos processos, actividades de controlo e interacção com utilizadores. De seguida, apresenta-se um resumo das mais importantes funcionalidades desta tecnologia: [Jur06]

- Descrição da lógica do processo de negócio através da composição de serviços.
- Composição de processos maiores a partir de outros processos mais pequenos ou serviços.
- Manipulação de invocações síncronas e assíncronas em serviços e gerir respostas que sejam recebidas após longos períodos de tempo.
- Invocação de operações em sequência ou paralelo.
- Compensação de actividades concluídas em caso de falha.

- Manter múltiplas actividades transaccionais de longa duração.
- Resumir actividades interrompidas ou falhadas para minimizar trabalho.
- Envio, recepção e reencaminhamento de mensagens para os processos e actividades respectivas.
- Estabelecer relação de pedidos dentro e entre processos de negócio.
- Agendar actividades a serem executadas e definir ordem de execução.
- Executar actividades em paralelo definindo o padrão de agregação.
- Estruturar processos em agrupamentos.
- Interação dos processos de negócio com os utilizadores, através de actividades de notificação e decisão.

### 2.1.2 Exemplo de um Processo de Negócio

A definição de um processo de negócio em BPEL na sua linguagem nativa em XML é caracterizada em 3 grupos: parceiros, variáveis e lógica de orquestração. O esquema apresentado representa o código fonte de um processo de negócio síncrono em BPEL, implementado num ficheiro com a extensão .bpel, que apenas recebe um parâmetro de entrada de um cliente e devolve uma resposta.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<process name="Exemplo"
    targetNamespace="http://xmlns.up.pt/Exemplo"
    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
    xmlns:client="http://xmlns.up.pt/Exemplo"
    xmlns:ora="http://schemas.oracle.com/xpath/extension"
    xmlns:orcl="http://www.oracle.com/XSL/Transform/java/
oracle.tip.pc.services.functions.ExtFunc"
    xmlns:xp20="http://www.oracle.com/XSL/Transform/java/
oracle.tip.pc.services.functions.Xpath20"
    xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
    xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
    xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/
business-process/">

    <!-- Partner Links -->
```

```

    <partnerLinks>
        <partnerLink name="client" partnerLinkType="client:Exemplo"
myRole="ExemploProvider"/>
    </partnerLinks>

    <!-- Variáveis -->
    <variables>
        <variable name="inputVariable"
messageType="client:ExemploRequestMessage"/>
        <variable name="outputVariable"
messageType="client:ExemploResponseMessage"/>
    </variables>

    <!-- Lógica de orquestração -->
    <sequence name="main">
        <receive name="receiveInput" partnerLink="client"
portType="client:Exemplo" operation="process" variable="inputVariable"
createInstance="yes"/>
        <reply name="replyOutput" partnerLink="client"
portType="client:Exemplo" operation="process"
variable="outputVariable"/>
    </sequence>
</process>

```

No início do processo são declarados todas as definições e extensões que são utilizadas. São declaradas as definições dos parceiros (Partner Links) e serviços que serão invocados pelo processo.

Em seguida, são declaradas as variáveis usadas e respectivos tipos. Estes podem ser agrupados em três conjuntos: tipos de dados nativos XML Schema, elementos definidos pelo utilizador e elementos do tipo de mensagens usadas para interagir com outros serviços. Neste exemplo, são declaradas duas variáveis que constituem os parâmetros de entrada e saída do processo, sendo a sua definição do tipo mensagem.

Por fim, temos a sequência de operações que compõem a lógica de orquestração do processo de negócio. Este exemplo é constituído por duas actividades executadas sequencialmente. A actividade recebe recebe os parâmetros de entrada do cliente na variável `inputVariable`. A actividade de reply retorna ao cliente o resultado na variável `outputVariable`.

Os processos BPEL são expostos como web services, logo é necessário a definição do serviço através de um ficheiro WSDL. Este ficheiro contém a definição da interface para esta operação e todos os tipos de mensagens trocados. Para este exemplo é apresentado uma possível definição do seu WSDL:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Exemplo"
    targetNamespace="http://xmlns.up.pt/Exemplo"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:client="http://xmlns.up.pt/Exemplo"
    xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link">

    <!-- Definição de tipos -->
    <types>
        <schema xmlns="http://www.w3.org/2001/XMLSchema">
            <import namespace="http://xmlns.up.pt/Exemplo" schemaLocation="Exemplo.xsd"/>
        </schema>
    </types>

    <!-- Definição das mensagens -->
    <message name="ExemploRequestMessage">
        <part name="payload" element="client:ExemploProcessRequest"/>
    </message>
    <message name="ExemploResponseMessage">
        <part name="payload" element="client:ExemploProcessResponse"/>
    </message>

    <!-- Definição das operações -->
    <portType name="Exemplo">
        <operation name="process">
            <input message="client:ExemploRequestMessage" />
            <output message="client:ExemploResponseMessage" />
        </operation>
    </portType>

    <!-- Definição dos parceiros de ligação -->
    <plnk:partnerLinkType name="Exemplo">
        <plnk:role name="ExemploProvider">
```



```
<plnk:portType name="client:Exemplo"/>
</plnk:role>
</plnk:partnerLinkType>
</definitions>
```

O ficheiro WSDL de um processo BPEL é constituído por quatro grupos. No primeiro grupo é feita a definição de tipos, podendo ser definidos directamente ou como neste exemplo, importando um ficheiro XML Schema onde se encontram especificados os tipos. No segundo, temos a definição dos tipos de mensagens usadas na comunicação e manipuladas no decurso do processo. Posteriormente, são especificadas as operações e as propriedades das ligações. Por fim, os tipos de parceiros de ligação do processo.

### 2.1.3 Aplicações e servidores disponíveis

Os servidores de BPEL proporcionam um ambiente de registo e execução de processos de negócio. Devido ao facto do BPEL estar fortemente relacionado com Web Services e as mais modernas plataformas que o suportam, Java Enterprise Edition (JEE) e Microsoft .NET, os servidores encontram-se actualmente desenvolvidos nestes ambientes. Os mais importantes servidores comerciais encontram-se na lista a seguir:

- Oracle BPEL Process Manager
- Microsoft BizTalk
- IBM WebSphere
- BEA WebLogic
- Sun Java Integration Suite for Java Enterprise Suite

Existem também soluções de código aberto dentro das quais se destacam:

- Red Hat JBoss Business Process Manager
- Apache Orchestration Director Engine
- ActiveBPEL Engine
- FiveSight Process Execution Engine PXE

Existem diversas aplicações de design e desenvolvimento gráficas, algumas das quais, incluídas com os servidores:

- Oracle JDeveloper BPEL Designer
- Eclipse BPEL Designer
- IBM WebSphere Studio Application Developer Integration Edition
- iGrafx BPEL
- Itp Process Modeler for Microsoft Visio

Na realização deste projecto foram utilizadas as ferramentas da Oracle, Oracle BPEL Process Manager e Oracle JDeveloper BPEL Designer.

## 2.2 Oracle BPEL Process Manager

A escolha dentro de todos os servidores de BPEL disponíveis para a realização deste projecto foi o Oracle BPEL Process Manager. As razões desta escolha prendem-se com o facto de já estarem implementados processos de negócio no Sistema de Informação. O Oracle BPEL Process Manager é um servidor aplicacional para implantação, execução e gestão de processos definidos em BPEL. Desenvolvido em Java, integra-se no Oracle Application Server através de Oracle Containers for J2EE (OC4J). A Oracle também disponibilizou diferentes versões que integram perfeitamente com outros servidores, como o da JBoss, Sun e IBM. As ferramentas disponibilizadas que fazem parte do pacote permitem desenhar, processar e administrar processos de negócio, orientados a serviços de forma rápida e intuitiva. Para além disso, integra-se muito bem com outras ferramentas da Oracle, com as suas vantagens e qualidades já bastante reconhecidas pelo mercado.

A lista a seguir apresenta as principais vantagens de usar Oracle BPEL Process Manager, descritas em [\[Jur06\]](#):

- Suporte de todos os standards BPEL desde a última especificação ws-bpel 2.0 até ao descontinuado BPEL4WS. Possibilitam a integração de arquitecturas orientadas a serviços nos diversos ambientes e sistemas empresariais.
- Inclusão de extensões proprietárias para melhor suporte de desenvolvimento desde a interacção com humanos e a debugging de processos.
- Integração com diversos servidores aplicacionais incluindo o Oracle Application Server, IBM WebSphere, BEA WebLogic e JBoss jBPM e diversas base de dados tais como Oracle, SQL Server, DB2, etc.
- Disponibilização de ferramentas de gestão e administração de processos de negócio em tempo real, incluindo visualização de estatísticas, detalhes de fluxo de actividade, desempenho dos processos, histórico de acções, entre outros...

- Inclusão de ferramentas de desenvolvimento de processos de negócio, o Oracle JDeveloper BPEL Designer e aplicações de debugging tais como Oracle BPEL Console e Worklist Application.
- Inclusão do sistema Dehydration para melhorar o desempenho de longos processos e a correlação de mensagens assíncronas.
- Disponibilização de uma API em Java que permitem administrar e gerir os vários processos BPEL do servidor.
- Integração com componentes de Java Enterprise Edition, tais como Enterprise Java Beans (EJB), Java Message Service (JMS), Web Services Invocation Framework (WSIF), entre outros.
- Suporte de várias versões do mesmo processo, permitindo que estes possam evoluir e que diversas instâncias a correr no sistema, possam terminar o seu processamento.
- Suporte para clustering permite que em caso de falha ou excesso de carga os processos possam ser realocados para outro servidor.

### 2.2.1 Arquitectura do Oracle BPEL Process Manager

A figura 2.4 mostra arquitectura do Oracle BPEL Process Manager constituída por quatro componentes principais: Servidor, Consola de gestão, Desenho de processos BPEL e Base de Dados. [dRM08]

#### 2.2.1.1 Desenho de Processos

O JDeveloper BPEL Designer permite o desenvolvimento gráfico de processos BPEL sem recurso à escrita directa em código. Por sua vez, permite arrastar actividades para o processo. No entanto, é igualmente possível editar directamente o código fonte destes processos, já que usa o formato XML nativo do BPEL. Permite, através de assistentes de criação, adicionar parceiros e localizar serviços, tais como: definição de tarefas a utilizadores, ligações a base de dados, etc, bem como adicionar funções, cópias de valores, construção de expressões. Permite colocar o processo directamente no servidor, facilitando o desenvolvimento e manutenção.

A figura 2.5 apresenta o aspecto do ecrã principal do Oracle BPEL Process Designer, integrado no Oracle JDeveloper. É possível visualizar, ao centro, a definição do processo de negócio BPEL mais simples, que apenas recebe e envia parâmetros a um cliente, quando é invocado. Na barra de actividades apresentada à direita, é possível identificar todas as actividades que podem compor a definição de um processo de negócio. Nas opções apresentadas em baixo, é possível alternar entre a apresentação gráfica do processo, o seu

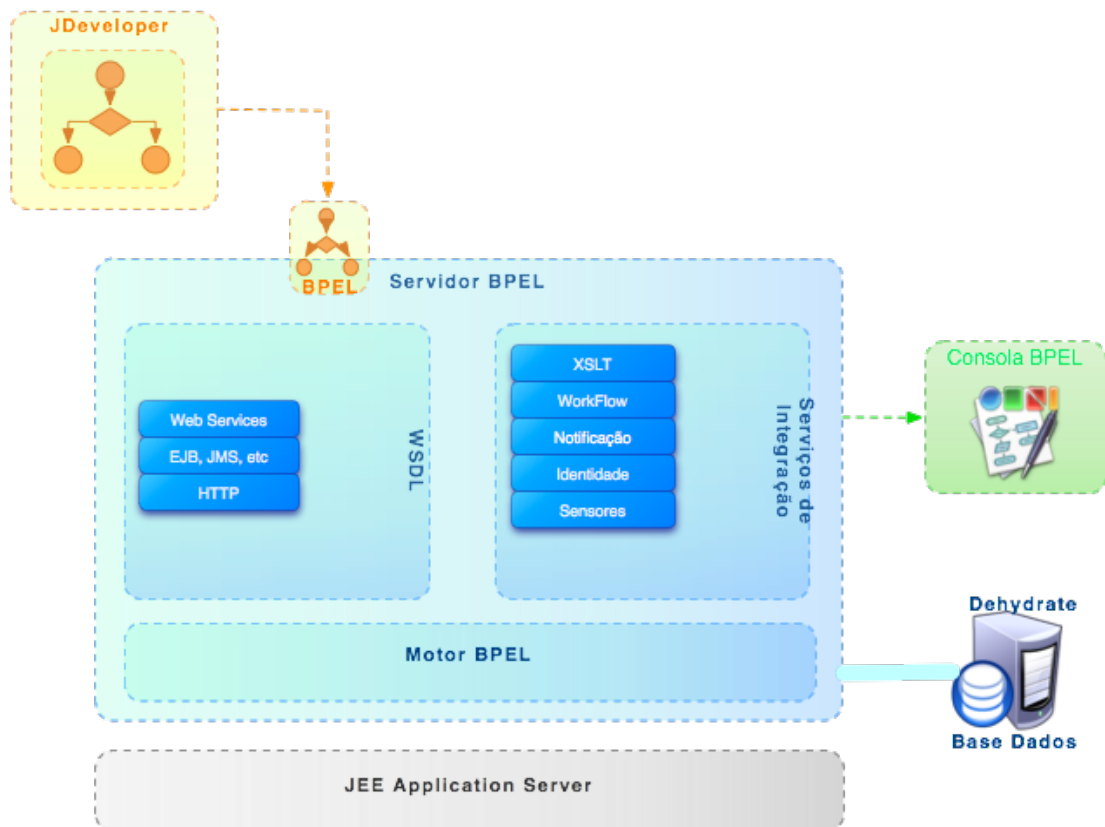


Figura 2.4: Arquitectura do Oracle BPEL Process Manager

código fonte e o histórico de alterações efectuadas. Para iniciar a definição do processo é necessário arrastar as actividades para o desenho do processo. A criação de algumas actividades, são auxiliadas por assistentes mais intuitivos que auxiliam a definição das suas propriedades principais como, por exemplo, a definição das ligações aos serviços externos.

### 2.2.1.2 Servidor BPEL

O servidor de BPEL é composto por um núcleo de processamento, componentes WSDL e serviços de integração. O núcleo de processamento é responsável pelo registo e processamento dos processos de negócio. Desenvolvido em Java, corre na plataforma Java Enterprise Edition (JEE) integrando-se com Oracle Containers for Java (OC4J) no servidor Oracle Application Server. Também suporta outros servidores, tais como, JBoss AS, BEA Weblogic e IBM WebSphere. Disponibiliza uma vasta API em Java que possibilita a implementação de ferramentas de administração e configuração de processos como o BPEL Console e o Worklist Application. [Jur06]

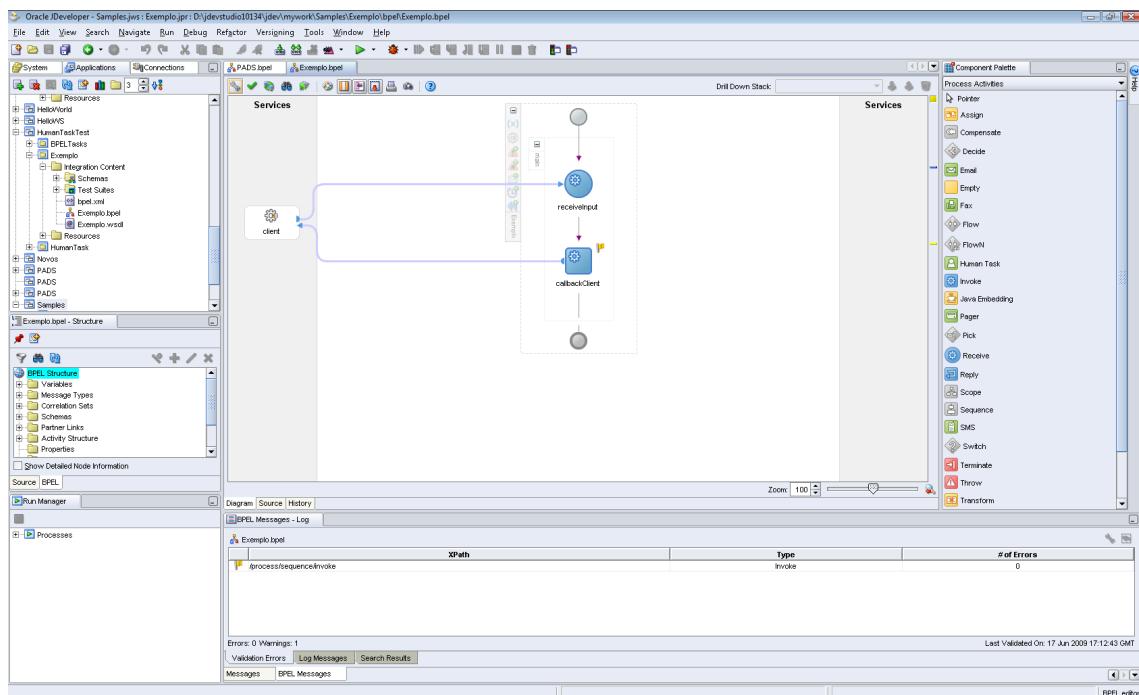


Figura 2.5: Oracle JDeveloper

Os componentes WSDL são responsáveis pela comunicação de processos BPEL, incluindo acesso a serviços por processos BPEL (parceiros). Apesar da especificação do BPEL apenas suportar web services, o Oracle BPEL Process Manager suporta outros protocolos além de SOAP, úteis em situações em que necessário aceder a aplicações ou sistemas já existentes destacando-se Enterprise Java Beans (EJB), Remote Method Invocation (RMI), Java Message Service (JMS), entre outros.

Os serviços de integração são constituídos por diversos componentes nos quais se destacam:

- XSLT - proporciona transformações complexas de XML.
- Workflow - fornece suporte para interações humanas tanto de tarefas como de decisões.
- Identidade - é uma camada que corre em cima da infra-estrutura do Oracle BPEL Process Manager, que fornece serviços de autenticação e autorização de utilizadores, pesquisa de propriedades do utilizador, funções, membros de grupos e privilégios. O serviço de identidade suporta os seguintes tipos de plug-in, tal como definido em [Ken07]:
  - JAZN - ficheiro de configuração em xml onde estão registados todas as propriedades dos utilizadores.

- LDAP - acesso a servidor de LDAP.
- Repositório personalizado - definição de um repositório personalizado. Cada configuração tem associado um dos tipos de serviço: Identidade - fornece a descrição dos utilizadores; Autenticação - fornece autenticação de utilizadores; autorização - fornece acesso a grupos, funções, autorizações e privilégios.
- Sensores - permitem que actividades, variáveis e falhas sejam monitorizadas durante a execução de processos BPEL. Úteis para reporte, integração e controlo de erros. Existem três tipos de sensores: [Ken07]
  - Database - Sensores do tipo database permitem guardar dados na base de dados do BPEL, acedidos através da API disponibilizada em JAVA.
  - JMS Queue/Topic - JMS Queue/Topic escrevem neste serviço de mensagens.
  - Custom - permite definir uma classe em java que implementa a interface Data-Publisher.

#### 2.2.1.3 Consola BPEL

A consola BPEL é uma aplicação integrada no Oracle BPEL Process Manager, com uma interface web que permite administrar e controlar processos BPEL. As principais características incluem:

- Gestão de processos (registar e apagar) e de instâncias (inicializar e apagar)
- Visualização dos processos activos e finalizados
- Visualização do conteúdo de sensores para uma determinada instância
- Acesso a histórico de informações
- Configuração do BPEL Process Manager

A figura 2.6 ilustra um exemplo de utilização da consola na visualização gráfica do fluxo corrente de actividades de um dado processo.

A worklist Application, representado na figura 2.7, é uma aplicação web de gestão de notificações e interacção com humanos. Permite que utilizadores se autenticuem usando o serviço de identificação e aceder às suas tarefas e notificações, pendentes ou não, e ainda tarefas que foram iniciadas por eles nos vários processos BPEL. Isto permite de uma forma rápida, fácil e intuitiva responder, cancelar ou apagar uma tarefa, influenciando um determinado processo de negócio.

## Arquitetura Orientada a Serviços

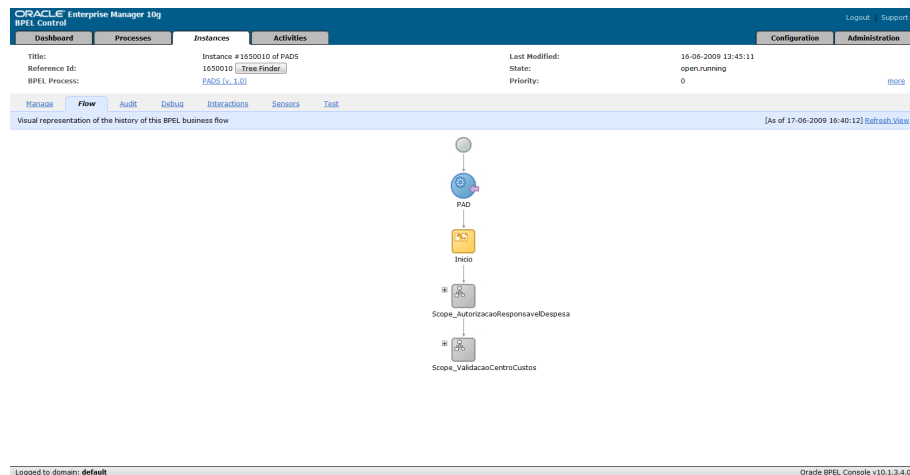


Figura 2.6: Oracle BPEL Console

The screenshot displays the Oracle BPM Worklist interface. The top navigation bar includes tabs for Home, Reports, Preferences, and Logout. The main content area shows a table of tasks with columns for Task Number, Title, Priority, Assigned Users, Assigned Groups, State, Created Date, and Expiration Date. The table contains two rows of tasks. The sidebar on the left includes sections for My Tasks (Inbox), Work Queues, Standard Views, My Views, Proxy Work Queues, and Delegated Views. A status bar at the bottom indicates the page was refreshed on Jun 17, 2009 4:14 PM.

Task Number	Title	Priority	Assigned Users	Assigned Groups	State	Created Date	Expiration Date	Actions
112975	Realização da Despesa pelo Centro de Custos Pad 1691	3	248271		Assigned	Jun 9, 2009 11:49 AM		-- Select an Action --
112989	Autorização da Despesa pelo Centro de Custos Pad 1692	3	248271		Assigned	Jun 16, 2009 1:50 PM		-- Select an Action --

Figura 2.7: Oracle Worklist

### 2.2.1.4 Base de Dados

O BPEL Process Manager suporta várias base de dados, para além da Oracle, das quais se destacam IBM DB2 e Microsoft SQL Server.

Uma importante funcionalidade, é o sistema dehydration e hydration. Como foi referido anteriormente, processos BPEL podem ser de longa duração, visto que chamadas a parceiros podem não ser respondidas de forma imediata, estes tem de ficar constantemente à espera de uma resposta para poder avançar o seu fluxo. Dehydration permite guardar todo o estado de um processo directamente na base de dados, libertando os recursos no servidor até que seja recebida uma resposta. Após recebida a resposta, todas as informações são restauradas da base de dados (Hydration), continuando novamente o seu normal processamento. [Jur06]

### 2.2.2 Conclusão

A decisão da escolha de utilização do Oracle BPEL Process Manager na implementação deveu-se ao facto de ser um requisito do projecto. Além disso, já se encontravam criados outros processos, o processo de Trouble Tickets, e já existia integração parcial com o SIGARRA. Mesmo assim, o oracle BPEL Process Manager é sem sombra de dúvidas, o mais completo e poderoso servidor applicacional existente. Concorrentes directos, que possuíam ferramentas comparáveis foram adquiridos pela Oracle como o caso da BEA. [dRM08]



## Capítulo 3

# Especificação do módulo

Este capítulo apresenta a especificação do módulo de pedidos de autorização de despesa (PAD). Pretende-se descrever todo o processo de negócio. Será feita uma descrição, identificados os actores, requisitos, apresentado o modelo de dados e todo o fluxo que compõe este processo.

### 3.1 Descrição

Pedidos de Autorização de Despesa (PAD) é um processo de negócio responsável por enviar pelas diversas entidades, solicitações para efectuar despesas ao longo de um percurso desde a sua criação até à sua finalização, cancelamento ou rejeição.

Fazendo uma visão geral, este processo começa pela criação do PAD, pelo autor da despesa que, pode ser o responsável por esta. São definidos os vários parâmetros do PAD, entre os quais se destacam o tipo de despesa (normal, reembolso, fundo de maneo, transferência interna), moeda, escolha do centro de custo e produtos/serviços requisitados. Em seguida, passa pela validação do responsável da despesa caso este não seja o autor. É feito o pedido de validação ao centro de custo associado. Atravessa o processo de cabimento executado na contabilidade. É realizado a validação do conselho administrativo. É neste momento, que é autorizada a despesa. Há a entrega de facturas, passando ao registo de terceiros. Por fim, há a integração da ordem de pagamento e autorização deste.

O processo de pedido de autorização de despesa é um processo complexo. A figura [3.1](#) apresenta uma versão simplificada do processo. Em anexo [A.0.1](#) encontra-se a versão completa. Pode ser dividido em várias fases: edição, validações, cabimento, validação administrativa, compromisso, inserção de facturas, registo de terceiros e pagamento. As próximas secções pormenorizam cada uma das fases.

### **3.1.1 Edição**

O sub-processo de edição consiste na notificação ao autor para que o PAD seja editado. Desta forma, o autor pode alterar o seu pedido e submetê-lo para avaliação, prosseguindo no fluxo ou cancelá-lo. Esta fase ocorre caso uma das validações não seja autorizada, ou não for possível efectuar o cabimento. A edição do PAD obriga que o processo volte ao início. A figura 3.2, apresenta o esquema:

### **3.1.2 Validações**

A fase de validações é separada em 2 momentos: validação do responsável da despesa e validação do centro de custo. Caso algum destes falhe é necessário ir para o sub-processo de edição. A figura 3.3 mostra a interação entre os sub-processos.

#### **3.1.2.1 Validação do responsável da despesa**

O processo de validação do responsável da despesa corresponde a uma notificação ao responsável a pedir autorização da despesa. Como mostra a figura 3.4, se o autor for responsável a autorização é automaticamente aceite.

#### **3.1.2.2 Validação do responsável do centro de custo**

A fase de autorização da despesa por parte do responsável centro de custo corresponde a uma notificação que à entidade que verifica e aceita a precisão da despesa, como mostra a figura 3.5. [San06]

### **3.1.3 Cabimento**

O cabimento é um acto pelo qual a autoridade competente acorda a realização de uma despesa concreta, por uma importância exacta ou estimada, procedendo-se à reserva (cativação) da totalidade ou uma parte da dotação correspondente. Assim sendo, este processo notifica a contabilidade informando que é necessário efectuar o cabimento. Caso seja necessário pode ser efectuado um pedido de assistência, sendo a pessoa em questão notificada. A figura 3.6 representa conceptualmente este processo. [San06]

### **3.1.4 Validação administrativa**

A autorização por parte da administração inclui a notificação à entidade administrativa responsável por aprovar ou rejeitar o pedido de despesa. A figura 3.7 representa o processo.

### **3.1.5 Compromisso**

No compromisso é enviado uma notificação ao autor da despesa informando que pode efectuar a despesa. A figura 3.8 representa o processo.

### **3.1.6 Inserção de facturas**

A inserção de facturas corresponde à fase onde o autor da despesa insere a factura em formato digital, tal como demonstrada na figura 3.9.

### **3.1.7 Registo de terceiros**

O registo de terceiros é composto por diversos passos. Numa primeira fase, há a espera de facturas em papel por parte dos serviços económico-financeiros, que verificam as facturas em termos legais e todos os itens legalmente exigidos. Caso o valor da factura diferir do valor proposto é enviado uma notificação à contabilidade para ser executado um ajuste do cabimento. Por fim, é enviado uma notificação à contabilidade para se proceder ao registo de terceiros que corresponde ao reconhecimento e registo da obrigação perante terceiros. A figura 3.10 sintetiza este processo: [San06]

### **3.1.8 Pagamento**

A fase de pagamento inicia-se com a integração da ordem de pagamento por parte da tesouraria. Isto é, é enviada uma notificação para registar as entidades às quais e vão efectuar o pagamento. Após esta actividade, caso existam despesas extraordinárias é necessário efectuar o sub-processo de registo de terceiros. Por fim, à a notificação final à tesouraria pedindo autorização para efectuar o pagamento. [San06]

## **3.2 Requisitos**

### **3.2.1 Funcionais**

Foram estipulados os seguintes requisitos funcionais para a nova versão deste módulo, tirando partido das vantagens da sua interligação com a tecnologia BPEL:

- Criação de um PAD
- Edição de um PAD
- Validação de um PAD
- Cancelamento de PAD
- Rejeição um PAD

- Visualização de PAD pendente
- Visualização de PAD finalizado
- Pesquisar um PAD usando diversos campos de selecção
- Efectuar cabimento
- Efectuar compromisso
- Ajustar cabimento/compromisso
- Registo de terceiros
- Integração da ordem de pagamento
- Autorização/Cancelamento da ordem de pagamento
- Inserção de facturas
- Espera de facturas em papel

### **3.2.2 Não funcionais**

De seguida, serão apresentados os principais requisitos não funcionais especificados para este sistema, que incluem algumas restrições ao seu funcionamento. Estes podem ser identificados nos seguintes agrupamentos:

- Requisitos de usabilidade:
  - As interfaces gráficas deverão ser intuitivas e a sua utilização deverá ser rápida e eficiente.
- Requisitos de serviços:
  - A disponibilização dos serviços deverá ser generalizada, de modo a funcionar correctamente com diferentes sistemas e tecnologias.
  - Deverá ainda ser considerada a inclusão de atributos de qualidade nos serviços tais como a segurança e a autenticação dos seus acessos.
- Requisitos de sistema:
  - O funcionamento deste projecto não deverá ficar dependente da estrutura de componentes que compõem o SIGARRA. Para isso, deverão ser eliminadas e generalizadas todas as ligações aos servidores, e às funcionalidades necessárias, de forma a possibilitar a sua futura integração em diversas instituições de ensino.

- O sistema deverá permitir facilmente futuras manutenções e actualizações.
- O sistema deve estar desenvolvido de forma que a alteração e construção de novos processos seja feita de forma fácil e intuitiva.

### 3.2.3 Casos de utilização

Dada a descrição apresentada e os requisitos funcionais, foram especificados os utilizadores envolventes com o sistema e os seus casos de utilização no novo módulo do Pedido Autorização de Despesa. Para isto, as figuras 3.12 e 3.13 apresentam o diagrama geral de pacotes UML dos casos de utilização definidos para este projecto. A figura 3.12 apresenta o pacote com os casos de utilizações comuns a todos os actores e a figura 3.13 os restantes.

#### 3.2.3.1 Actores

Os actores que interagem directamente com as funcionalidades deste módulo são utilizadores registados no sistema de informação do SIGARRA. Estes podem ser distinguidos pelas seguintes categorias:

- Autor da despesa - representa a entidade que criou um PAD. Faz o pedido pelo responsável da despesa.
- Responsável da despesa - entidade que é responsável pelo PAD.
- Responsável centro de custo - é uma entidade que lhe é atribuída uma divisão orçamental onde tem de prestar contas dos movimentos efectuados. [dEdUdP09]
- Conselho administrativo - cabe competências de gestão administrativa, patrimonial e financeira. [dR]
- Contabilidade - representa os serviços económico-financeiros. Prepararam e acompanham a execução do orçamento, organizam e apresentam o balanço e a demonstração de resultados e processam as receitas e despesas. [dEdUdP09]
- Terceiros - entidade que faz o reconhecimento e registo da obrigação perante terceiros.
- Tesouraria - entidade que executa os pagamentos, arrecada receitas e mantém actualizada a conta corrente das contas de depósito. [dEdUdP09]

#### 3.2.3.2 Pacotes

As funcionalidades do modulo de PAD estão organizadas em seis pacotes:

## Especificação do módulo

- Geral - pacote no módulo de PAD que agrupa as funcionalidades de consulta e pesquisa PADs pendentes ou finalizados com diferentes campos de selecção.
- Lançamento do pedido de autorização de despesa - pacote que agrupa as funcionalidades de criação e edição de PADs. A inserção de facturas está presente, bem como a funcionalidade de enviar um PAD para edição.
- Administração - pacote que inclui autorização, cancelamento e rejeição de PADs.
- Contabilidade - agrupa as funcionalidades de contabilidade. Estas incluem efectuar, ajustar e rejeitar cabimento/compromisso e editar detalhes contabilísticos.
- Terceiros - agrupa as funcionalidades de registo de terceiros.
- Tesouraria - agrupa as funcionalidades de ordens de pagamento. Estas incluem autorizar, rejeitar e integrar uma ordem de pagamento com possibilidade de despesas extras.

## Especificação do módulo

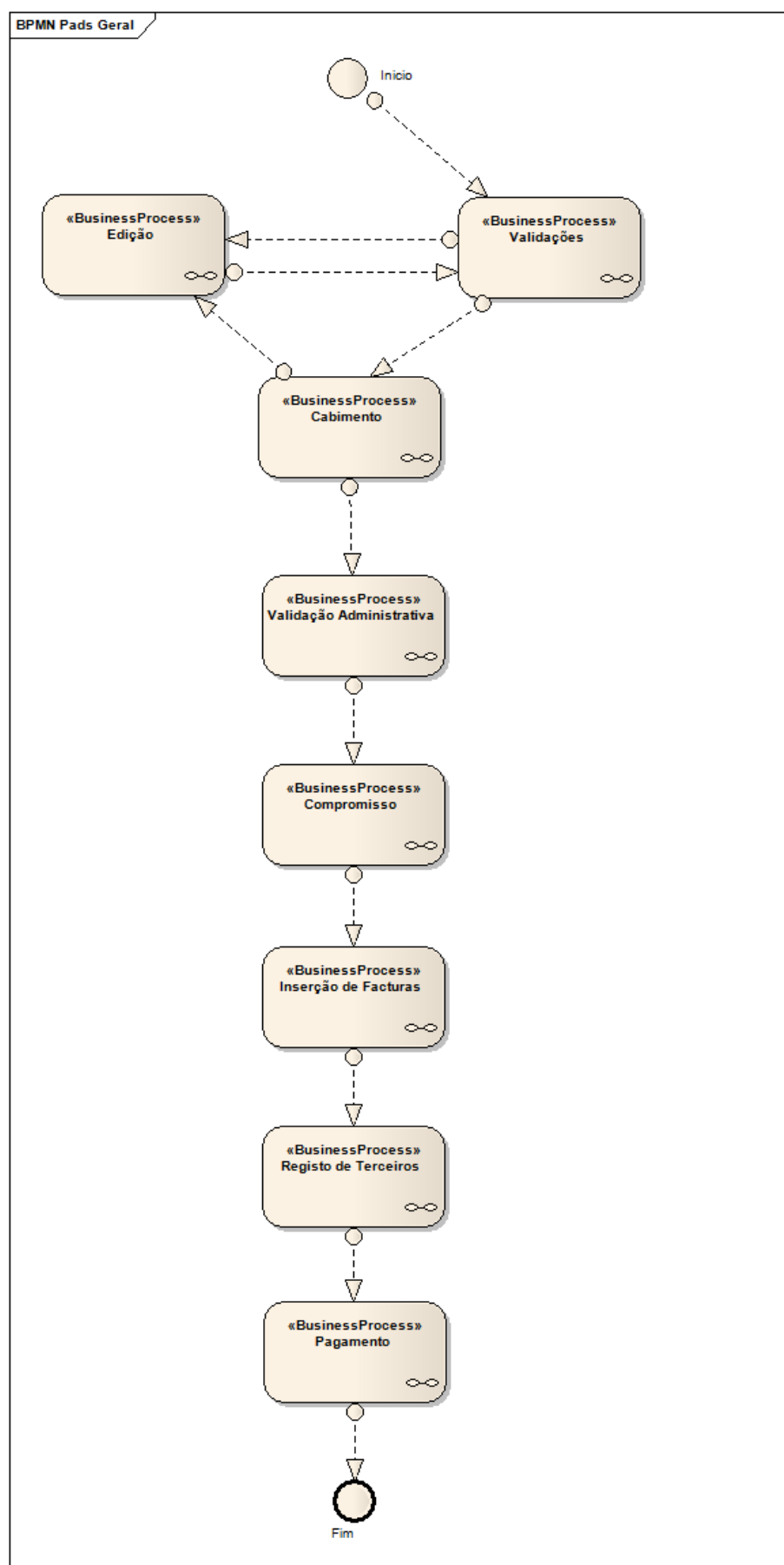


Figura 3.1: Processo de PAD<sup>28</sup>

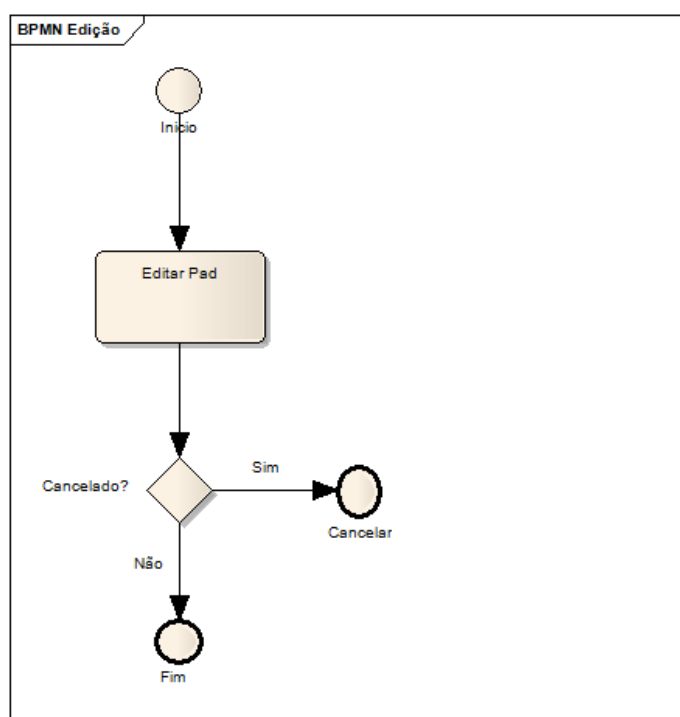


Figura 3.2: Fase: Edição



## Especificação do módulo

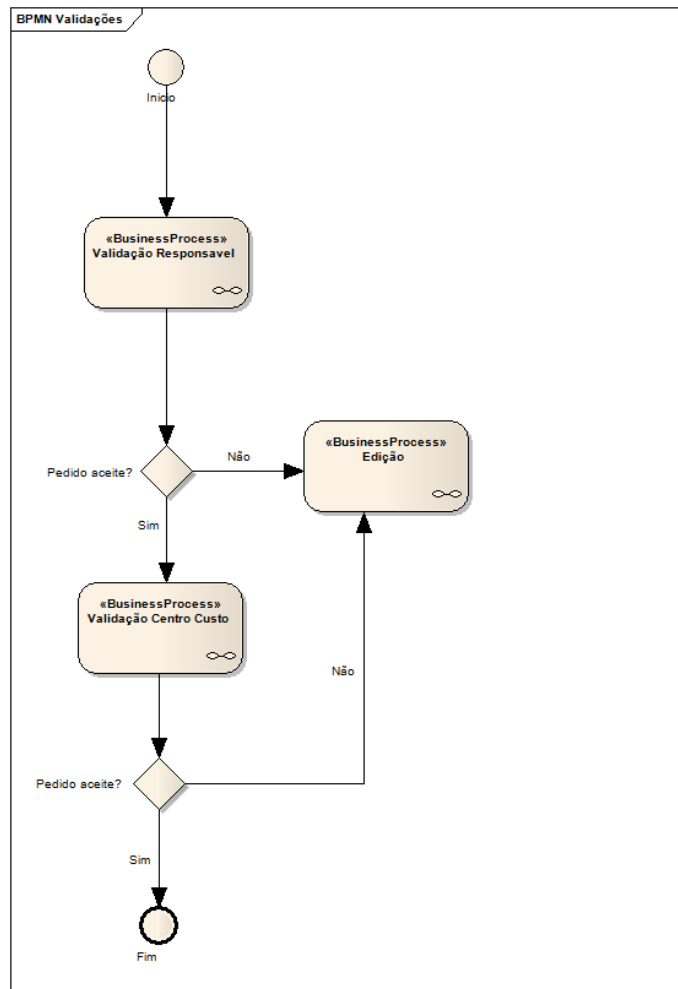


Figura 3.3: Validações

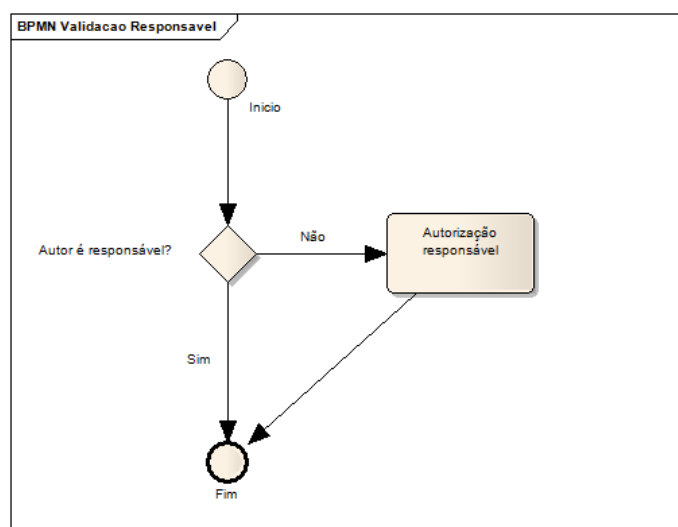


Figura 3.4: Fase: Validação do responsável da despesa

## Especificação do módulo

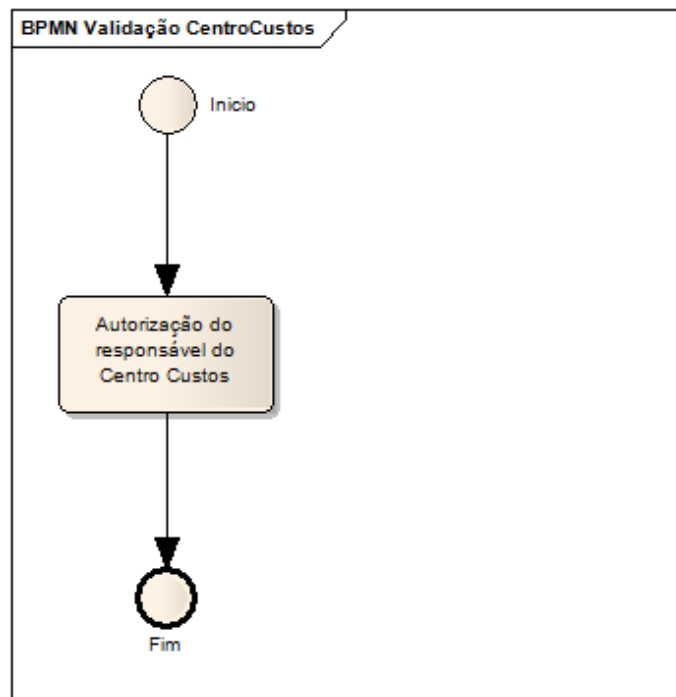


Figura 3.5: Fase: Validação do responsável do Centro de Custo

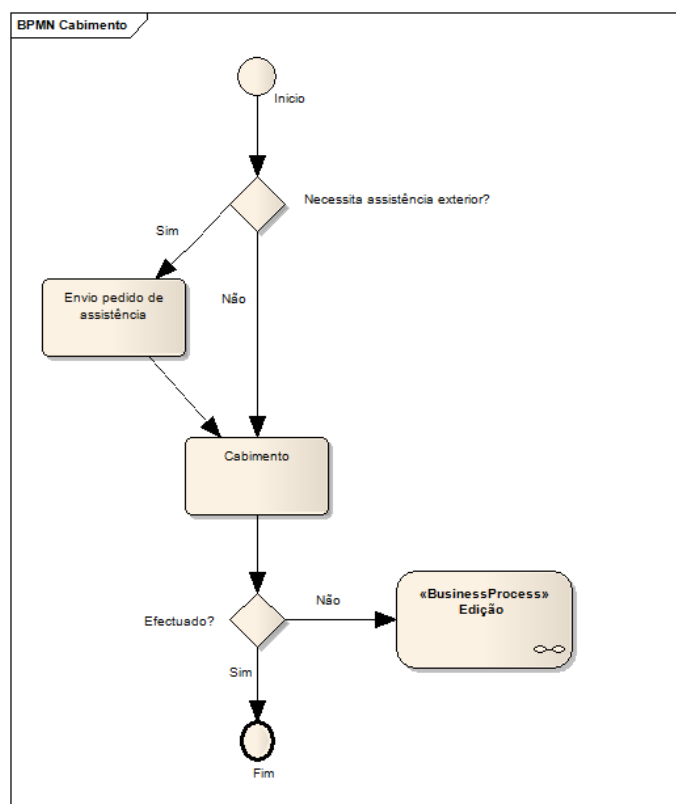


Figura 3.6: Fase: Cabimento

## Especificação do módulo

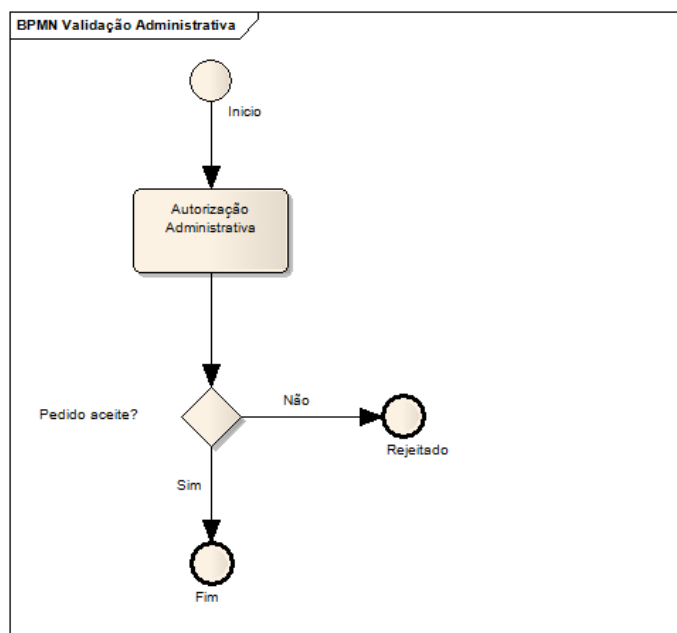


Figura 3.7: Fase: Validação Administrativa

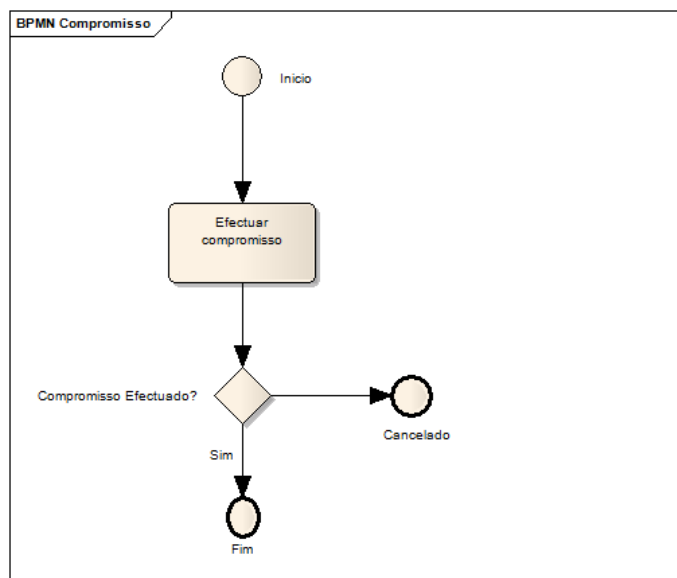


Figura 3.8: Fase: Compromisso

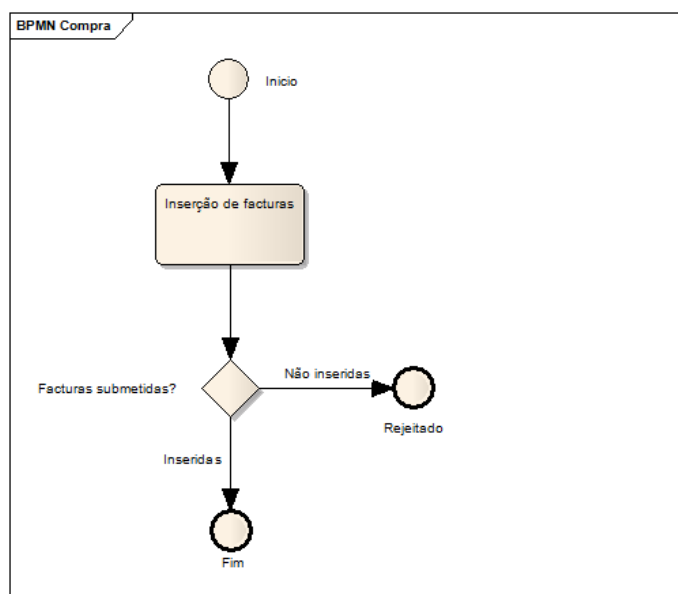


Figura 3.9: Fase: Inserção de Facturas

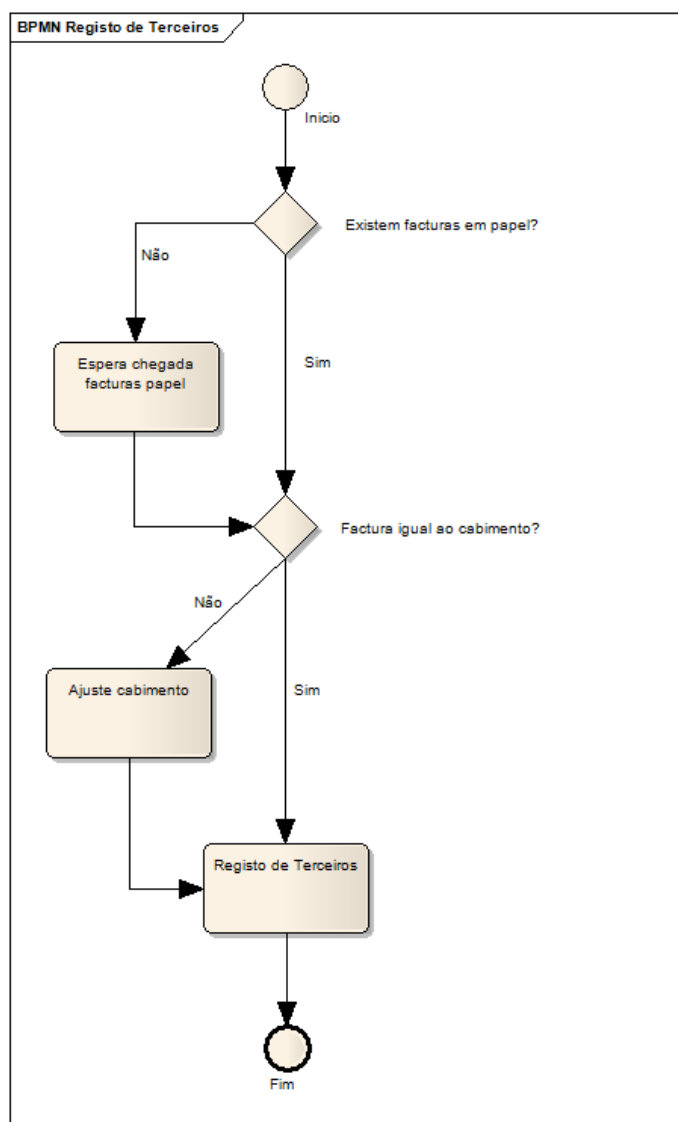


Figura 3.10: Fase: Registo de Terceiros

## Especificação do módulo

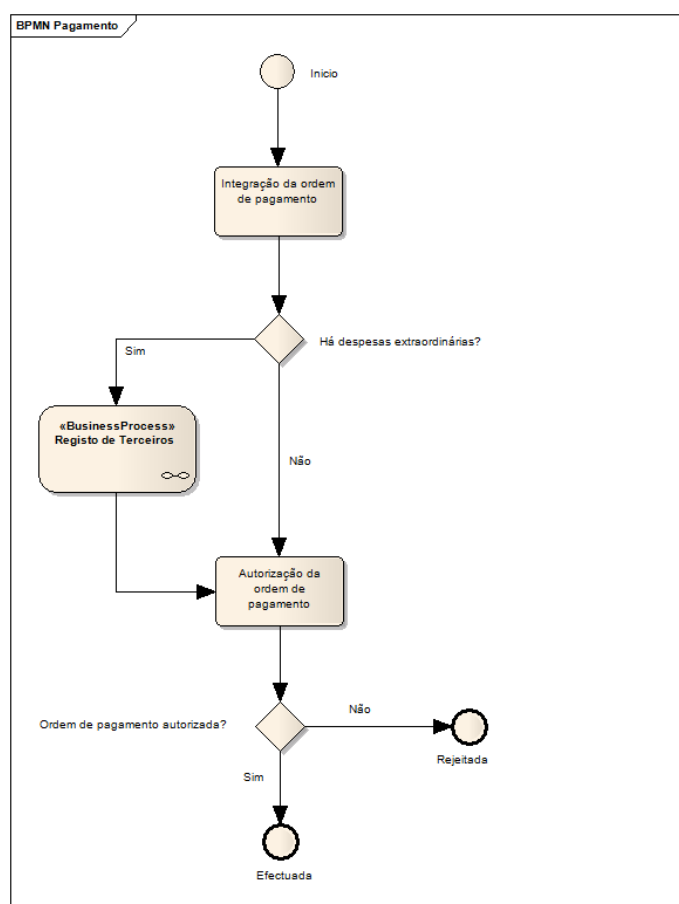


Figura 3.11: Fase: Pagamento

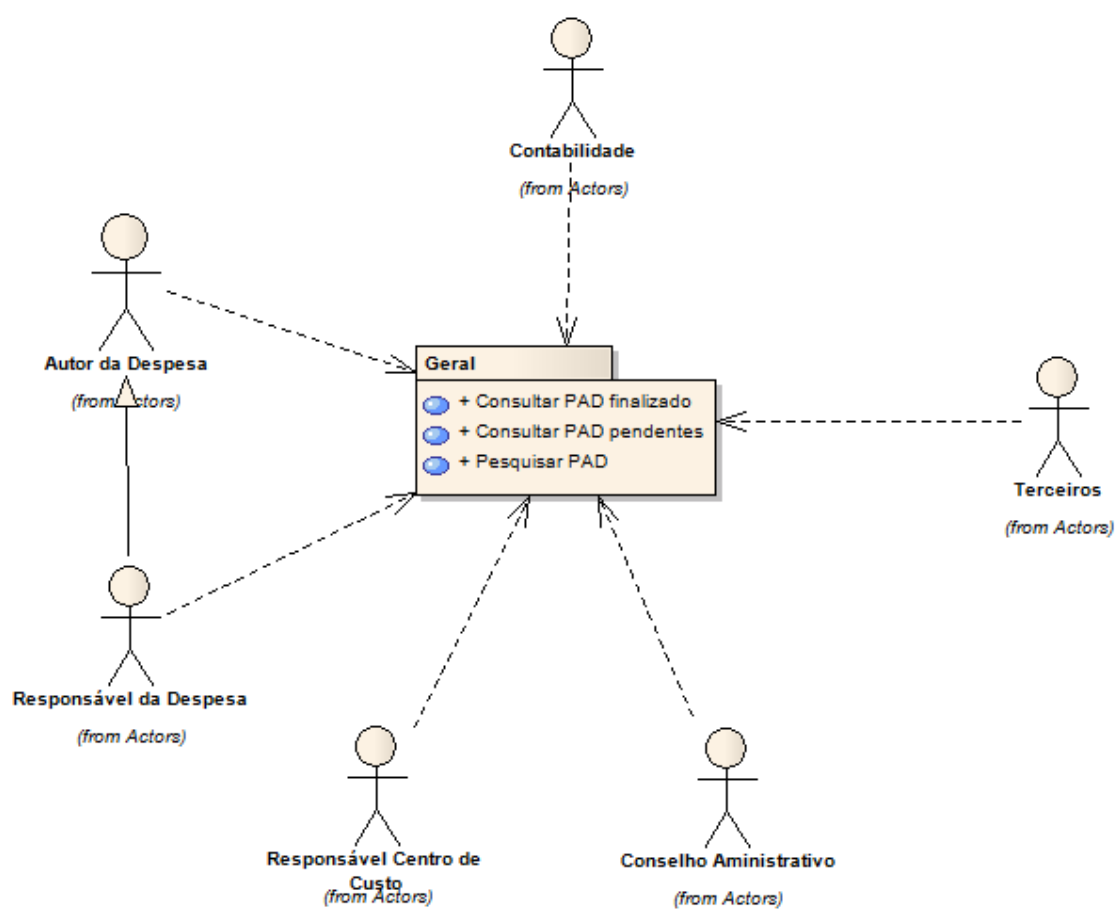


Figura 3.12: Casos de utilização gerais

## Especificação do módulo

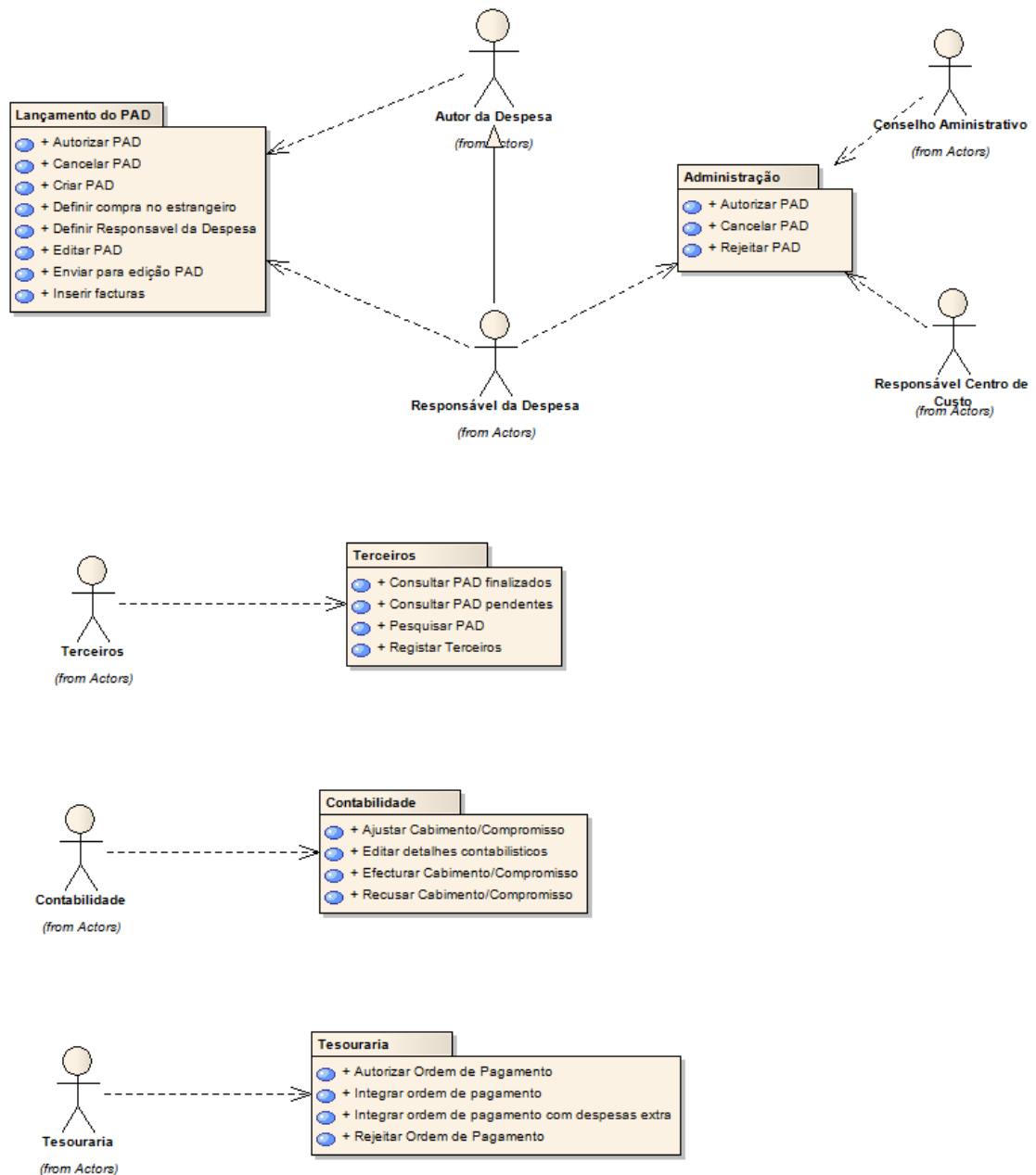


Figura 3.13: Casos de utilização por pacotes



## Capítulo 4

# Implementação

Após a especificação da solução proposta para o módulo, descrevem-se neste capítulo os detalhes da sua implementação, da gestão de ligações e da integração de componentes e tecnologias, de forma a cumprir os requisitos e objectivos deste projecto.

Serão apresentados nas seguintes secções, os detalhes da arquitectura lógica e física do sistema distribuído implementado, a lógica de negócio do módulo, terminando na estrutura e organização dos componentes do processo de negócio.

### 4.1 Arquitectura Lógica do Sistema

#### 4.1.1 Decomposição Horizontal

A criação do módulo de pedidos de autorização de despesa foi elaborado de forma a corresponder às especificações da arquitectura SIBPEL, enquadrando-se em três camadas lógicas hierárquicas: interface, lógica de negócio e acesso a dados. Estas camadas diferem de acordo com o nível de abstracção em que se encontram, com o tipo de informação com que interagem e com as ligações que utilizam. A representação da arquitectura horizontal do sistema é apresentada na figura [4.1](#).

##### 4.1.1.1 Interface

Esta camada engloba todos os componentes visíveis para o utilizador a que este pode aceder para interagir com o sistema, sendo deste modo, a única camada visível para o utilizador final. Foi aproveitada a camada de interface do módulo de pedidos de autorização de despesa já existente, que contém, todas as páginas e formulários de utilização e interacção. Da mesma forma, a camada de interface do gestor de tarefas que já se encontrava desenvolvida.

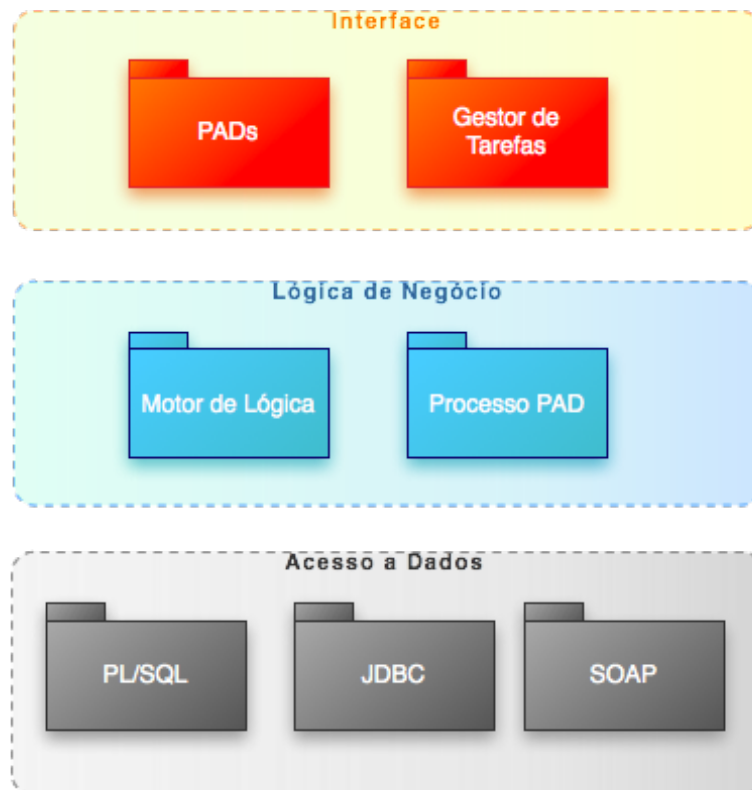


Figura 4.1: Decomposição Horizontal

#### 4.1.1.2 Lógica de Negócio

A camada da lógica de negócio tem como objectivos disponibilizar todo o motor para os módulos do sistema de informação. Neste projecto, as suas funcionalidades aplicam regras de validação, tratamento de dados, fluxos de actividades e disponibilização de serviços. Para isto, esta camada é principalmente composta pelos seguintes componentes:

- **Motor de Lógica** - É essencialmente composto por procedimentos e funções PL/SQL que definem as regras de negócio do novo módulo de Pedidos de Autorização de Despesa e a comunicação e interacção com a tecnologia BPEL e Serviços. Aqui se encontram implementados os pacotes de funcionalidades e regras, bem como os da arquitectura SIBPEL em termos de comunicação entre PL/SQL, BPEL e Web Services.
- **Processo Pedido de Autorização de despesa** - processo de negócio definido em BPEL que complementa as funcionalidades do módulo de SI implementado no Motor de Lógica. Foi igualmente sobre este componente, que a implementação deste projecto se apoiou bastante. Deste modo, todo o processo de negócio do módulo dos

Pedidos de Autorização de Despesa foi totalmente redefinido para BPEL, estando a sua interacção com o seu motor de lógica completamente funcional e sincronizado. Os processos de negócio BPEL comunicam com os restantes componentes e com a camada de dados através de mensagens SOAP e JDBC.

### 4.1.1.3 Acesso a Dados

Esta camada identifica os componentes necessários para interligar as duas camadas adjacentes, permitindo que a camada superior da Lógica de Negócio aceda aos dados presentes na Camada de Dados. Para isto, as comunicações entre ambas são realizadas utilizando três tecnologias diferentes:

- **PL/SQL** - Os comandos SQL são principalmente utilizados no acesso ao repositório do SIGARRA pelo Motor de Lógica. No entanto, através da disponibilização de procedimentos e funções em PL/SQL como serviços, os processos de negócio em BPEL utilizam igualmente esta tecnologia aliada ao SOAP para realizar o acesso tanto ao repositório do SIGARRA como ao do BPEL.
- **SOAP** - Esta tecnologia é principalmente utilizada na definição de mensagens de comunicação entre os dois componentes pertencentes à Lógica de Negócio. As mensagens são definidas numa estrutura XML que permite incluir os parâmetros necessários para invocar processos e serviços, bem como retornar os resultados por estes devolvidos. Quanto ao acesso a dados, esta tecnologia é essencialmente utilizada pelos Processos de Negócio em BPEL para aceder a procedimentos e funções em PL/SQL como serviços externos.
- **JDBC** - Este constitui o componente utilizado pela API em Java, usada na implementação da arquitectura SIBPEL, para acesso de dados sobre utilizadores pelo serviço de identidade e inserção de dados sobre processos pelos sensores no SIGARRA.

### 4.1.2 Decomposição Vertical

Nesta secção é apresentado um modelo mais funcional da arquitectura do sistema com base num diagrama de pacotes vertical, que permite visualizar as interacções entre os diversos componentes e módulos do sistema.

Visualizando o diagrama apresentado na figura 4.2, a todas as interfaces gráficas desenvolvidas está associado o respectivo motor de lógica que implementa as suas funcionalidades e aplica as regras necessárias. Quanto à lógica do novo módulo de Pedidos de Autorização de Despesa este comunicará com o seu processo de negócio em BPEL através da geração, envio e recepção de mensagens SOAP. Por outro lado, irá igualmente efectuar acessos e manipulações de dados directamente sobre o seu modelo relacional definido na base de dados do SIGARRA, onde acede igualmente o processo BPEL.

Relativamente à interface do Gestor de Tarefas acede às informações dos processos, instâncias e tarefas do BPEL através dos Web Services previamente implementados, que utilizam a API de Java fornecida de modo a aceder e controlar os dados contidos no repositório do servidor de BPEL. Os acessos a estes serviços são efectuados também através da troca de mensagens SOAP, que por sua vez acedem ao repositório de BPEL por JDBC. Os processos de negócio BPEL interagem igualmente com este repositório nomeadamente para implementar a funcionalidade Dehydration.

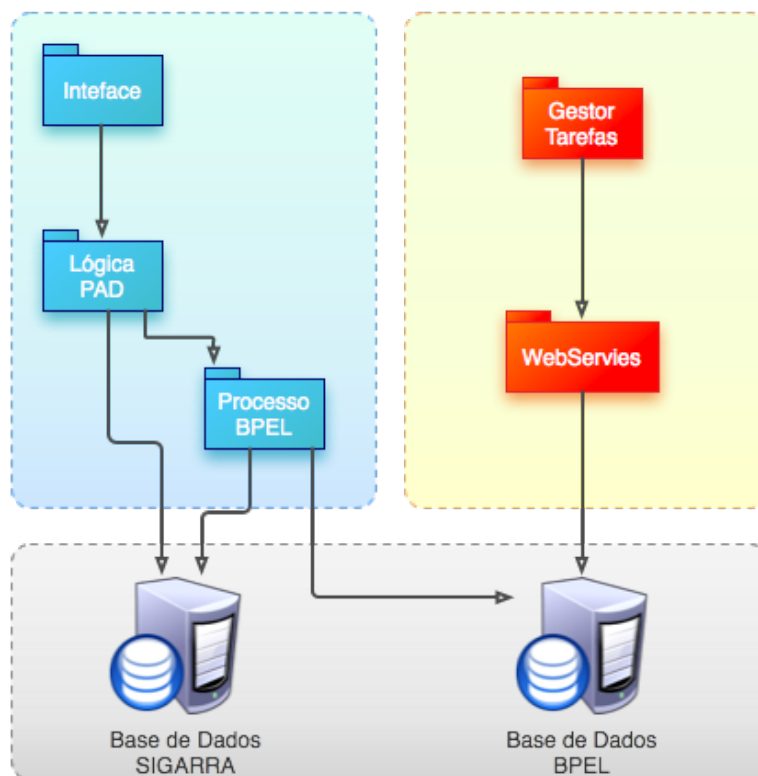


Figura 4.2: Decomposição Vertical

## 4.2 Arquitectura Física do Sistema

No que respeita à distribuição física dos componentes do sistema, esta permite capturar a topologia de hardware constituinte, os diversos componentes e módulos de software utilizados pelo sistema e as respectivas localizações. Para exemplificar a arquitectura física do sistema, foi elaborado um diagrama de distribuição que permite capturar toda a estrutura física da implementação, sendo este apresentado na figura 4.3.

Neste diagrama estão representados os componentes distribuídos por apenas dois servidores. No servidor de dados corre a base de dados que contém toda a infra-estrutura

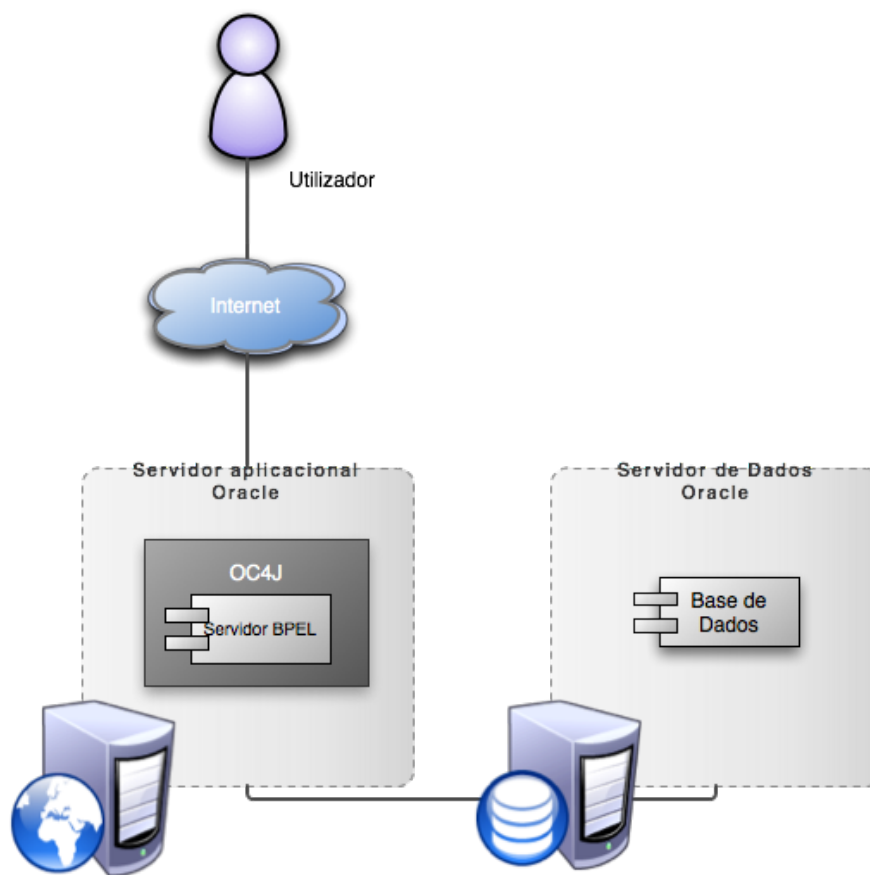


Figura 4.3: Arquitectura Física

do SIGARRA. O servidor aplicacional corresponde ao Oracle Application Server, uma variante do Apache HTTP Server. Este contém uma componente, o Oracle Containers for J2EE (OC4J), que suporta Java Enterprise Edition usado no BPEL Process Manager.

### 4.3 Sistema SIBPEL

A implementação do Sistema SIBPEL neste projecto representa a criação de toda a base necessária para integração e comunicação do Sistema de Informação com o Oracle BPEL Process Manager. Parte desta arquitectura já se encontrava desenvolvida. O Oracle BPEL Process Manager é composto por três componentes, Sensores, Serviço de identidade e Workflow como mostra a figura 4.4.

#### 4.3.1 Serviço de identidade

Tal como explicado no capítulo 2.2.1.2, o serviço de identidade permite acesso a serviços de autenticação e autorização de utilizadores, pesquisa de propriedades do utili-

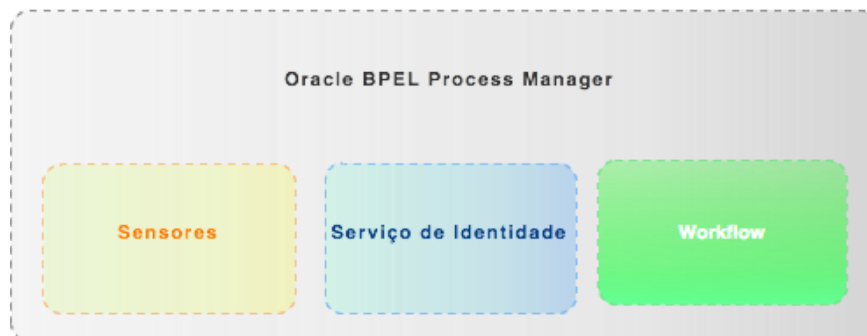


Figura 4.4: Arquitectura SIBPEL

zador, funções, membros de grupos e privilégios. Dos três serviços de plugin suportados foi utilizado o custom dado que é o único que se adequa à integração com o SIGARRA, visto tratar-se de acessos à base de dados. O plugin custom expõe uma interface em Java.

Na altura da realização deste projecto já se implementada parte da estrutura. As funcionalidades já criadas eram compostas pela identificação e autenticação de utilizadores. Desta forma, foi necessário rever toda a estrutura feita e acrescentar o suporte a grupos de utilizadores, sendo implementados outros serviços como acesso a informações e pesquisa de utilizadores. A tabela 4.1 apresenta as funções implementadas e respectiva descrição:

Tabela 4.1: Lista de operações implementadas no serviço de identidade

Operação	Descrição
lookupUser	Instância um utilizador, inicializando todas as suas informações sabendo o seu código
getUsers	Devolve todos os utilizadores
searchUsers	Pesquisa utilizadores a partir de um padrão
getUsersByPredicate	Devolve os utilizadores de um dado grupo

#### 4.3.2 Sensores

Os sensores permitem identificar alterações no estado de variáveis numa dada instância. Em todos os sub-processos é usado uma variável LogMessage responsável por guardar informações de estado e notificações. A esta variável está associado um sensor do tipo custom, referido no capítulo 2.2.1.2, efectua ligações JDBC à base de dados do SIGARRA actualizando informações do estado do processo. Na tabela 4.2 está sintetizado os atributos da variável LogMessage. No anexo A.3 encontra-se o modelo de dados.

Tabela 4.2: Lista de operações implementadas no serviço de identidade

Nome	Descrição
PAD_ID	Número de identificação do PAD
Estado	Estado do processo
Resultado	Resultado de uma tarefa
Observação	Observação efectuada pelo votante
Notificados	Elementos notificados para execução da tarefa
Votante	Elemento que respondeu à notificação

### 4.3.3 Workflow

O serviço de workflow é constituído por um conjunto de webservices que interagem com a API disponibilizada em Java, do Oracle BPEL Process Manager que disponibiliza acesso às notificações. Desta forma, o gestor de tarefas invoca estes webservices apresentando aos utilizadores as notificações deste.

## 4.4 Novo módulo - Pedidos de Autorização de Despesa

Após a implementação de todo o Sistema SIBPEL, foi iniciado o processo de migração do módulo de Pedidos de Autorização de Despesa na versão em Oracle Workflow para BPEL. Toda a construção e redefinição deste novo módulo foram apoiadas em todos os conceitos apresentados nos capítulos anteriores.

### 4.4.1 Definição dos Processos de Negócio BPEL

A definição dos processos de negócio BPEL para o novo módulo dos Pedidos de Autorização de Despesa, marcou o início da elaboração deste novo sistema. Analisando todo o fluxo de actividades [3.1](#), aliado à análise dos novos requisitos e casos de utilização propostos na especificação deste modulo, foi decidido separar o processo de PADs em vários sub-processos.

Nesta nova definição foram introduzidas novas actividades e conceitos em todo o fluxo. A diferença mais significativa é a repartição do processo em 9 sub-processos diferentes. A partição dos processos foi efectuada segundo a especificado no capítulo [3.1](#). Esta partição foi efectuada de modo a possibilitar a reutilização em futuros processos de negócio e tirar partido de todas as vantagens da tecnologia BPEL no cumprimento dos conceitos da Arquitectura SOA, nomeadamente na reutilização de serviços e na redução da complexidade do sistema.

Devido ao requisito não funcional do sistema [3.2.2](#), o processo principal deve ser feito de forma a que o desenvolvimento seja de forma fácil e rápida. Desta forma, todo o processo de retorno ao estado inicial caso algumas das validações não seja efectuada

foi colocado na lógica dos sub-processos. O processo principal tornou-se extremamente simples sendo que os sub-processos tornaram-se mais complexos.

Outra decisão de implementação corresponde ao tratamento de erros. Dado que toda a comunicação efectuada pelo BPEL, é feita por web services, estes estão sujeitos a falhas de comunicação. Desta forma, foi decidido que todas as chamadas devem ser refeitas até se conseguir. Cada tentativa é separada de um minuto.

De seguida são apresentados os sub-processos:

### 4.4.1.1 Editar PAD

O sub-processo de edição do PAD é composto por duas fases. Numa primeira fase, chama um web service que getPadAutor, que devolve o código do seu autor. Após ter sido recebido, procede-se à segunda fase: envio da tarefa para o autor. Este recebe a notificação, sendo que pode editar o PAD, e de seguida envia-lo para aprovação ou cancela-lo. A figura 4.5 mostra o desenho do processo em BPEL.

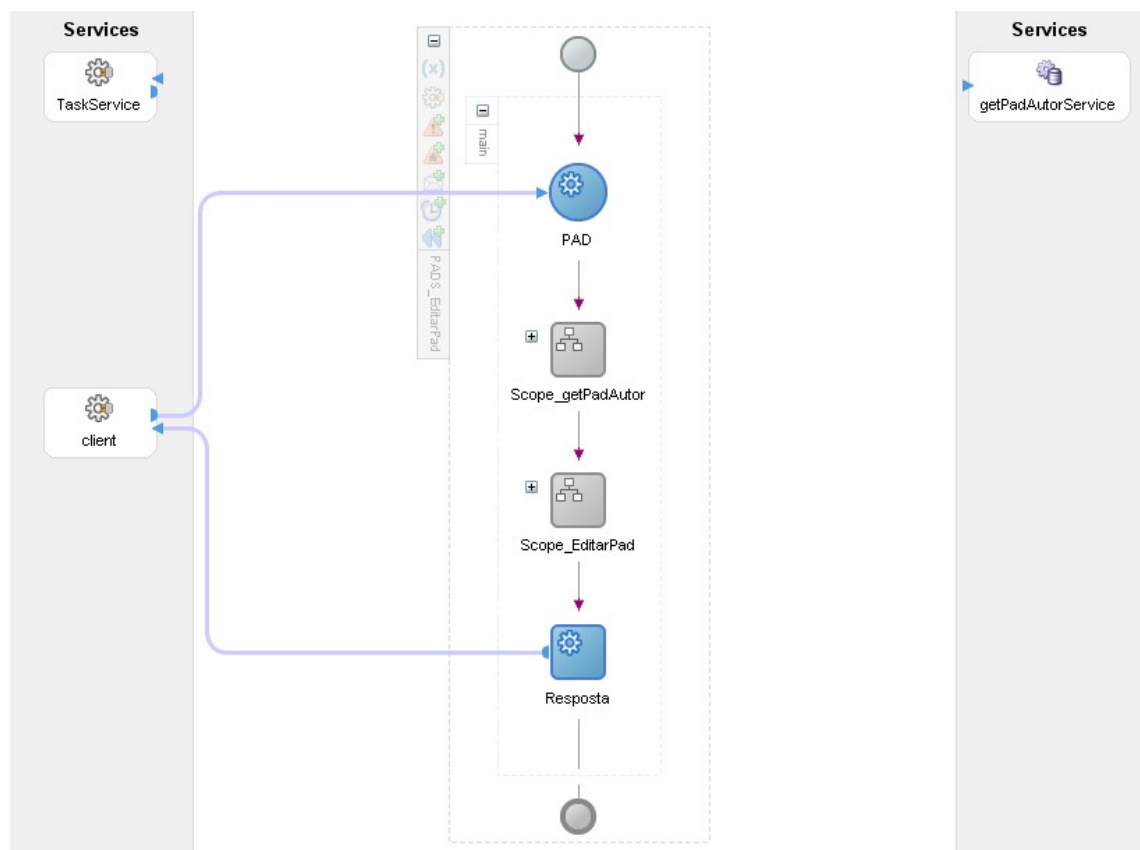


Figura 4.5: Edição do Pad



#### 4.4.1.2 Validação responsável da despesa

O sub-processo de validação por parte do responsável é composto por uma verificação que compara o autor com o responsável. Caso este seja o mesmo, a validação é automaticamente aprovada. Caso não seja, é enviada uma notificação de aprovação da despesa para o responsável da despesa. A figura 4.6 mostra o desenho do processo em BPEL.

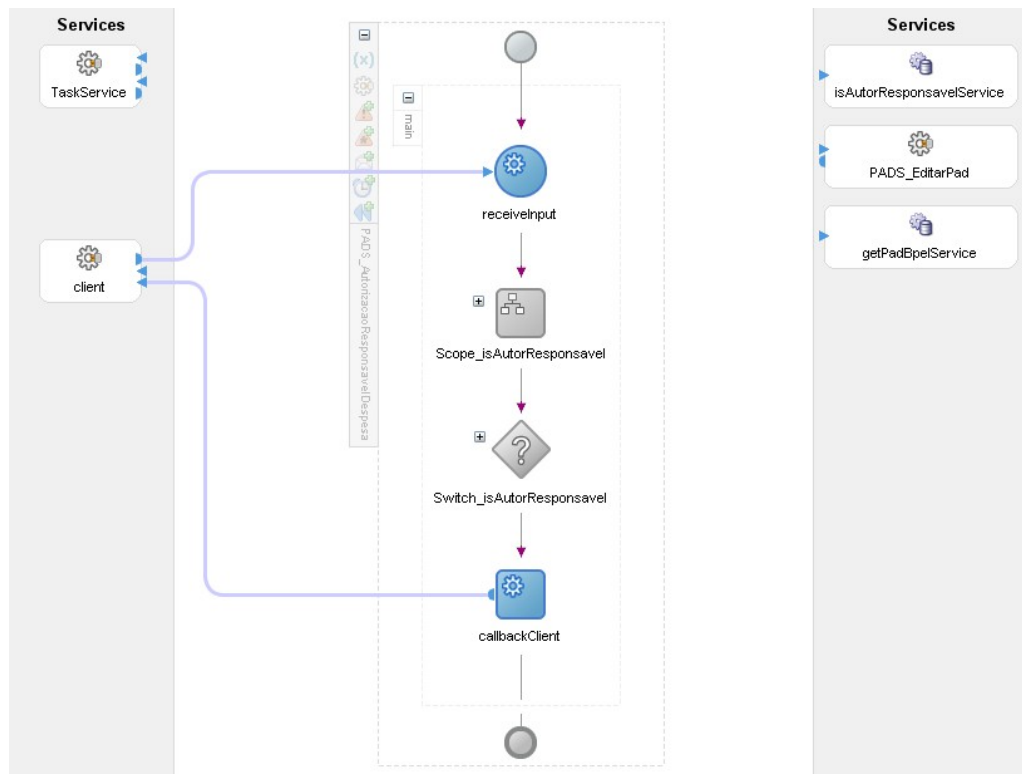


Figura 4.6: Validação responsável da despesa

#### 4.4.1.3 Validação do Centro de Custo

O sub-processo de validação por parte do responsável do centro de custo é enviado uma notificação de validação ao responsável do centro de custo. A figura 4.7 mostra o desenho do processo em BPEL.

#### 4.4.1.4 Cabimento

Neste sub-processo é enviado uma notificação à contabilidade para se efectuar cabimento. A figura 4.8 mostra o desenho do processo em BPEL.

## Implementação

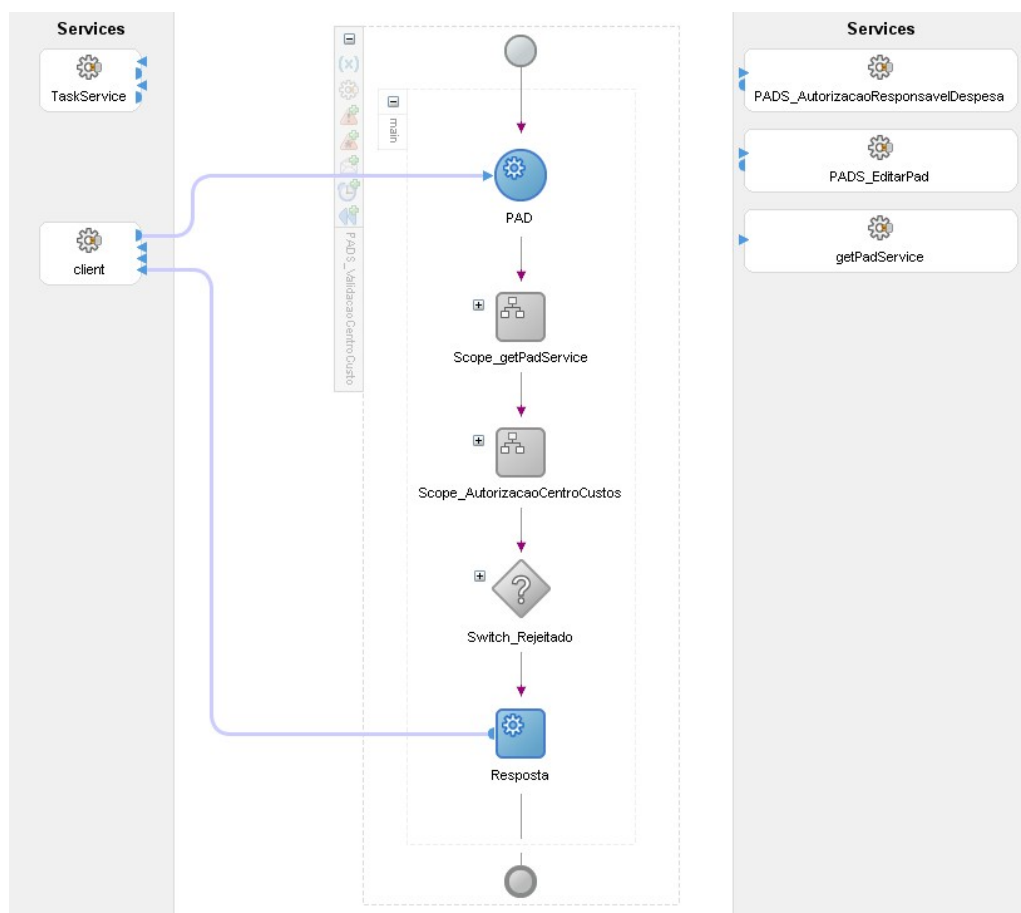


Figura 4.7: Validação do Centro de Custo

### 4.4.1.5 Validação Administrativa

No sub-processo de validação administrativa incorpora uma notificação ao conselho administrativo pedindo autorização para se efectuar o PAD. A figura 4.9 mostra o desenho do processo em BPEL.

### 4.4.1.6 Compromisso

No processo de compromisso é enviado uma notificação ao autor informando que pode efectuar a compra. Este pode efectuar a compra ou cancelá-la. A figura 4.10 mostra o desenho do processo em BPEL.

### 4.4.1.7 Inserção de facturas

O processo de inserção de facturas corresponde a uma tarefa: envio de uma notificação ao autor do PAD, a inserção online da digitalização da factura. A figura 4.20 sintetiza esse processo.

## Implementação

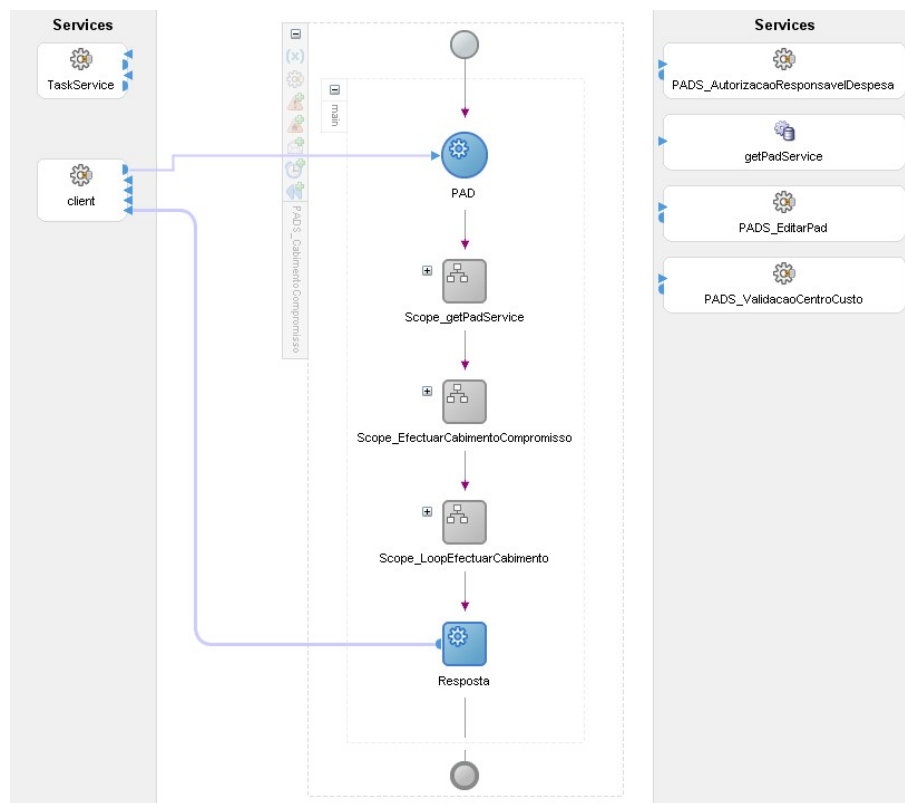


Figura 4.8: Cabimento

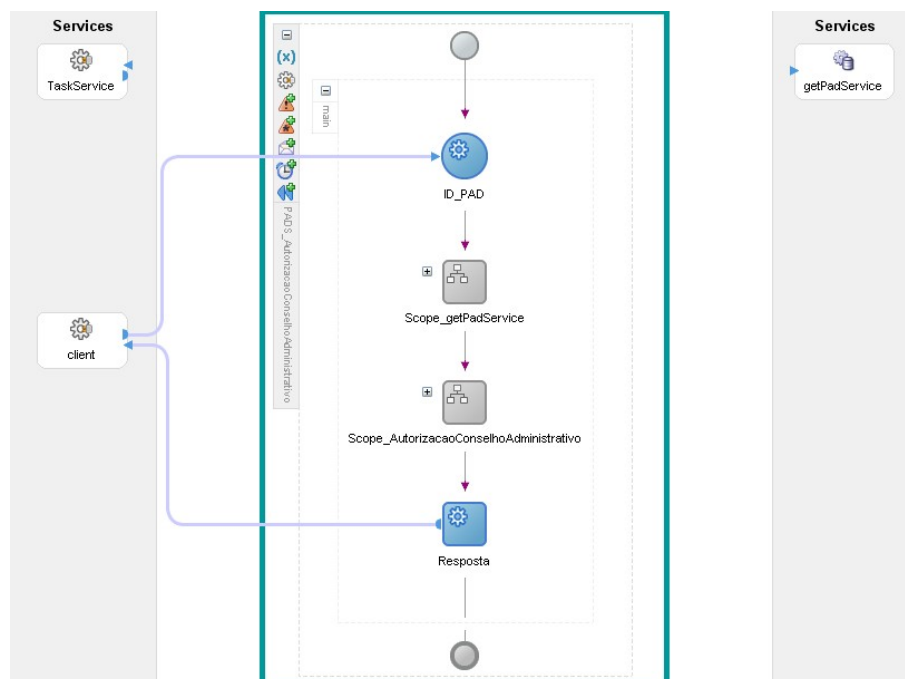


Figura 4.9: Validação Administrativa

## Implementação

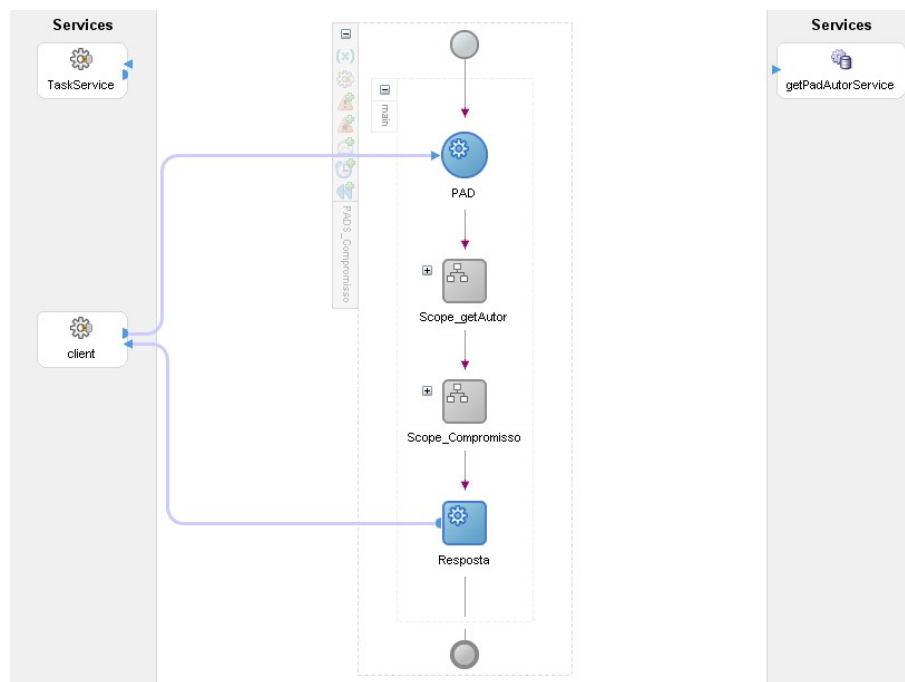


Figura 4.10: Compromisso

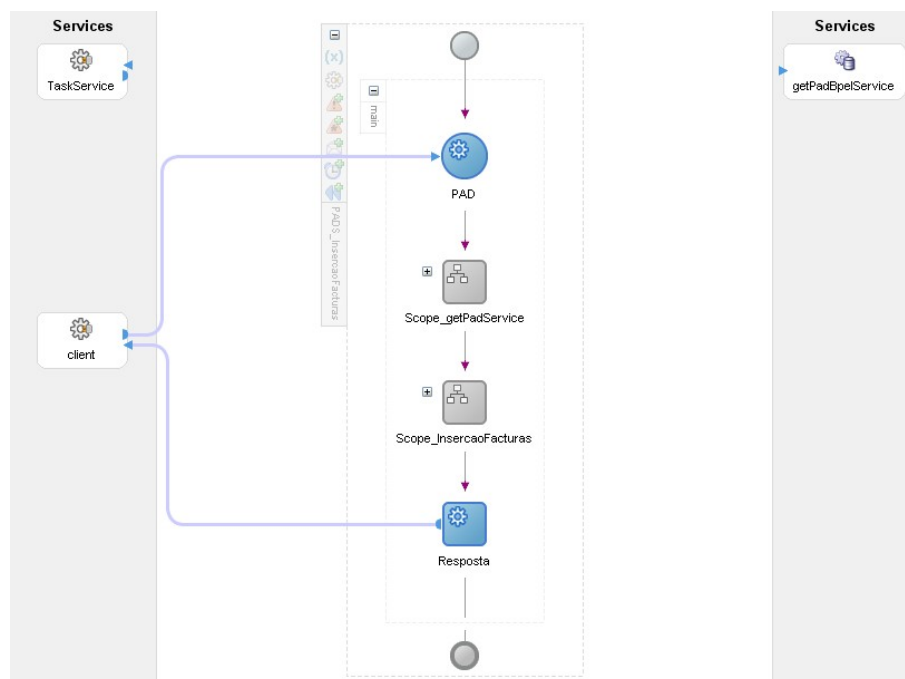


Figura 4.11: Inserção de facturas

#### 4.4.1.8 Registo de terceiros

O processo de registo de terceiros é composto por três fases: espera de facturas em papel, ajuste de cabimento e registo de terceiros. Na tarefa de espera de chegada de facturas em papel, é enviado uma notificação para confirmação da chegada da factura ao serviços económico-financeiros. Após recepção, é verificado se a factura excede o valor da proposta. Caso seja é enviado uma notificação à contabilidade para efectuar uma nova cabimentação. Por fim, é enviado uma notificação ao autor para efectuar o registo de terceiros.

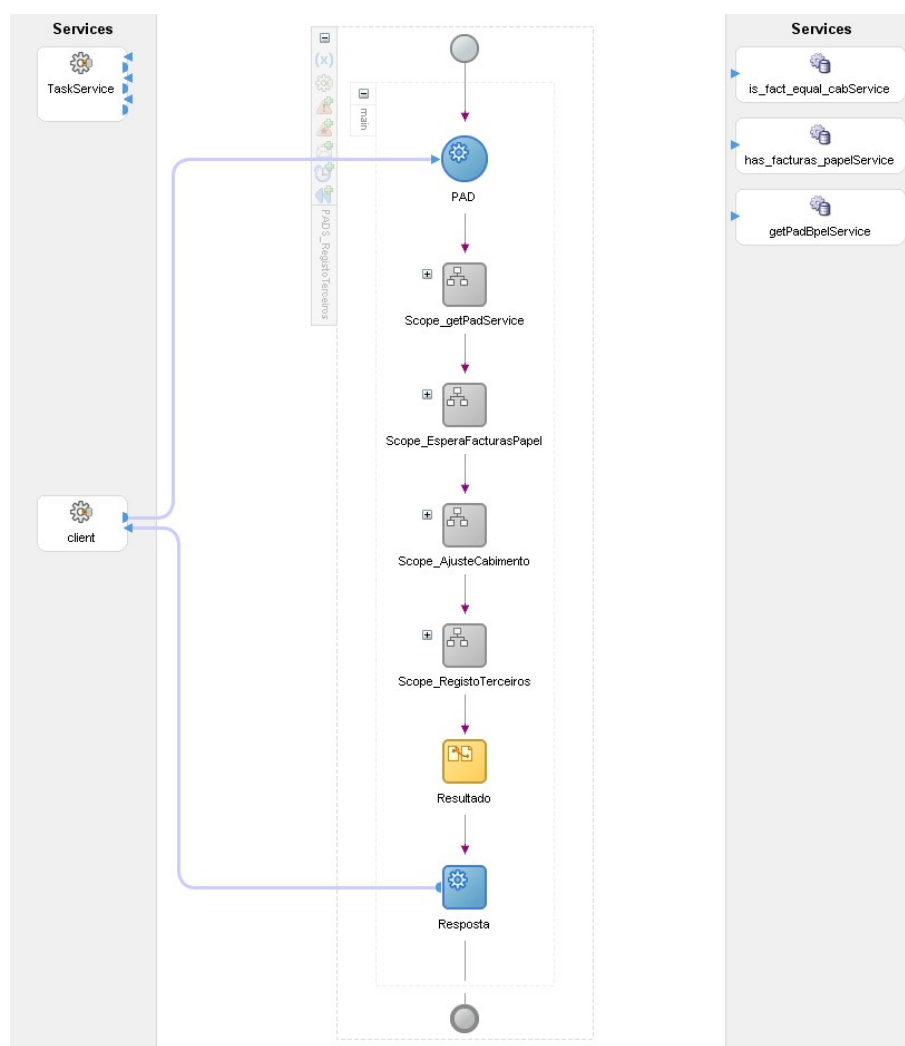


Figura 4.12: Registo de terceiros

#### 4.4.1.9 Pagamento

Tal como especificado no capítulo 3.1.8, o sub-processo de pagamento é composto por três fases. Na primeira fase há integração da ordem de pagamento. Caso existam despesas

extra superiores a um montante estabelecido, é efectuado o registo de terceiros. Por fim, procede-se à autorização da ordem de pagamento.

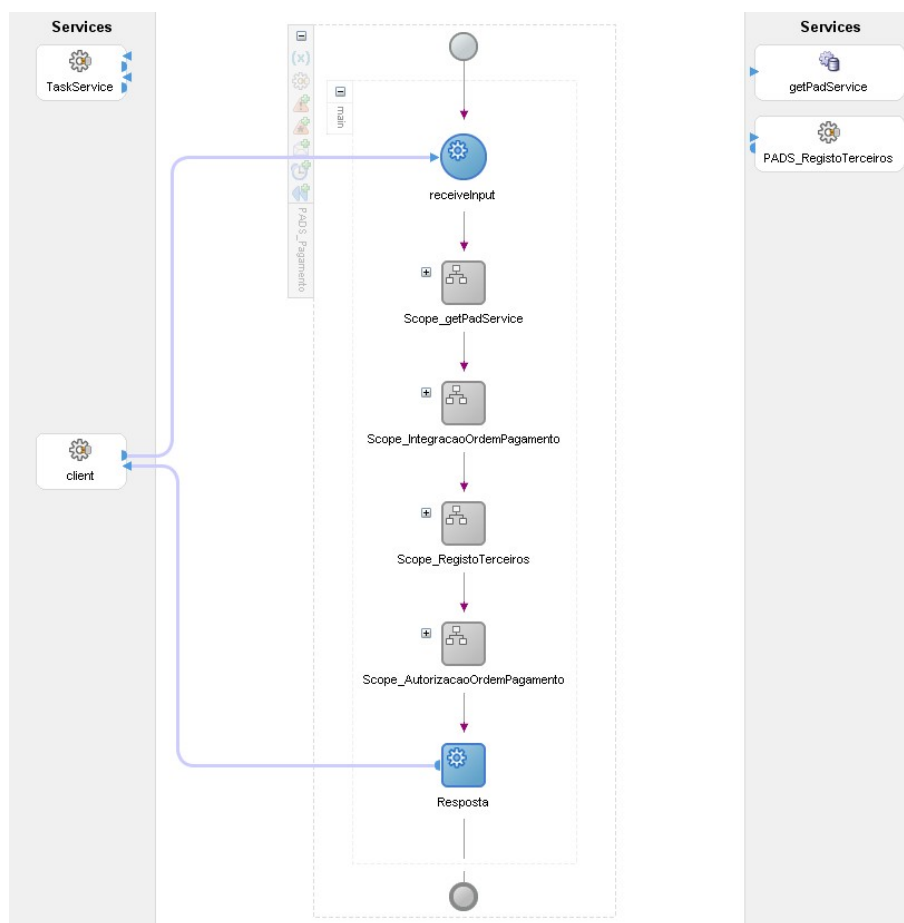


Figura 4.13: Validação do Centro de Custo

### 4.4.1.10 PAD

O processo PAD, corresponde ao processo principal. É responsável por invocar os sub-processos que desta forma constituem o processo de Pedido de Autorização de Despesa. Seguindo a especificação no capítulo 3.1, o processo inicia o seu percurso, começando pela validação pelo responsável da despesa e validação do centro de custos. Em seguida há o sub-processo cabimento por parte da contabilidade. Até esta fase o pedido pode ser rejeitado e editado para tentar validação, tal como mostra o fluxo completo em anexo A.0.1. Em seguida há a validação administrativa e o compromisso. Segue-se a inserção de facturas sendo verificado se a factura não excede o previsto. Caso ultrapasse é efectuado a validação do centro de custos e validação administrativa. Por fim, é chamado o sub-processo de pagamento. Todo este processo é apresentado na figura resumida 4.14.

## Implementação

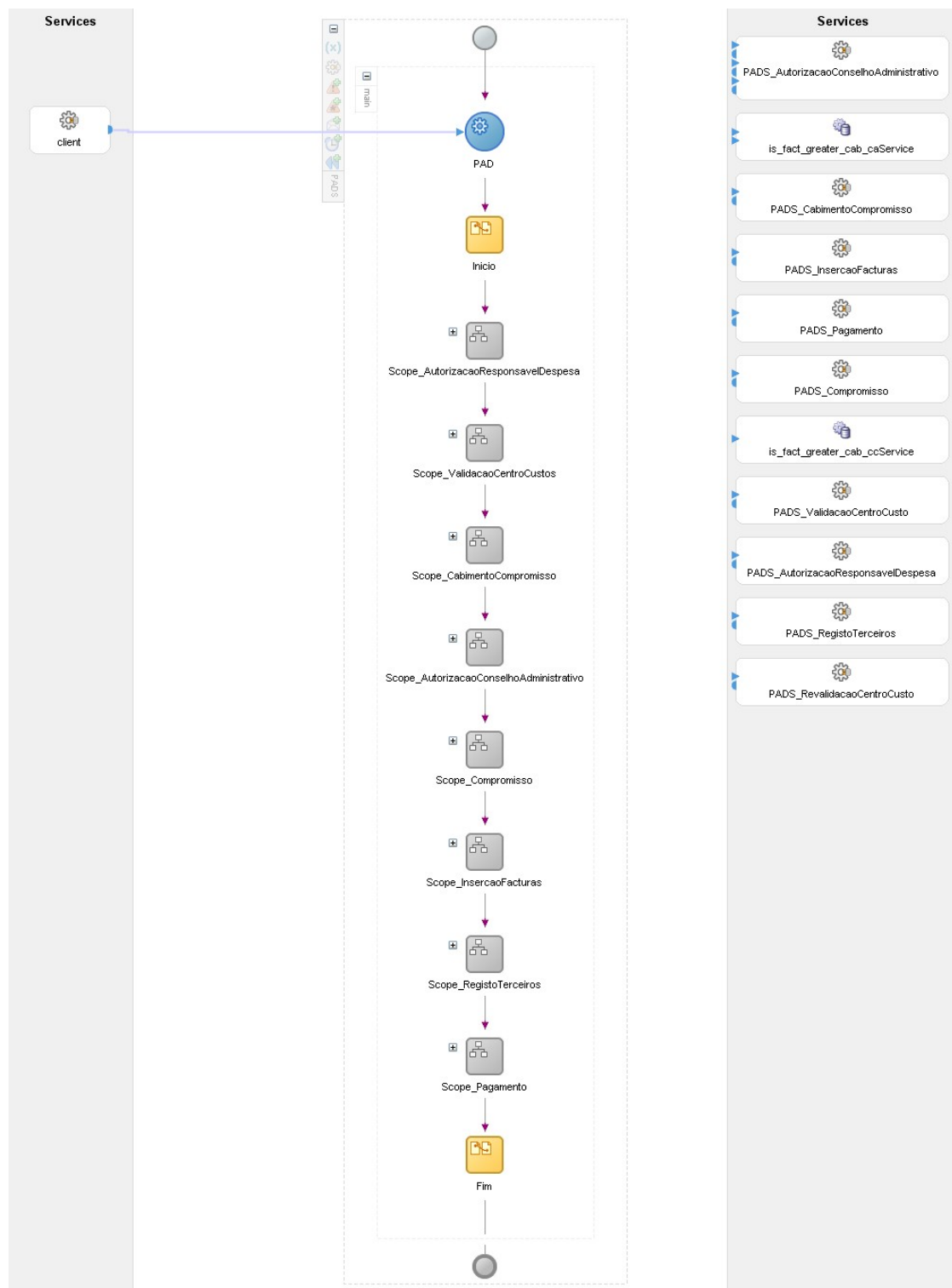


Figura 4.14: Pedido de Autorização de Despesa

#### 4.4.2 Modelo relacional de Dados

A nível do modelo relacional de dados deste novo módulo, este sofreu igualmente pequenas alterações comparativamente com a sua versão antiga em WorkFlow. Desta forma, tentou-se manter a futura conversão o mais simples e pouco evasiva. Foram incluídas no modelo todas as necessidades de sincronização com as instâncias BPEL, onde se destaca o estado de cada processo, os resultados e as entidades notificadas. Para isto, foi definido o modelo relacional de dados apresentado no anexo [A.0.2](#) para esta nova versão do módulo de Pedidos de Autorização de despesa.

#### 4.4.3 Motor de lógica

Todo o motor de lógica da versão em WorkFlow foi implementado em procedimentos e funções PL/SQL [[McL08](#)]. Desta forma, decidiu-se aproveitar todo o desenvolvimento já realizado. No entanto, o código não se encontrava em bem organizado e separado segundo um modelo lógico. Para isto, foram implementados basicamente cinco pacotes de procedimentos:

- **PADS\_BPEL:** Disponibiliza todas as funcionalidades e todo o motor de lógica necessário para o funcionamento do novo módulo. Aqui são ainda invocadas as operações do Sistema SIBPEL de forma a integrar e sincronizar a tecnologia BPEL com este módulo.
- **PADS\_UTIL:** Disponibiliza procedimentos e funções que, no geral, complementam todas as funcionalidades.
- **PADS\_GERAL:** Aqui incluem-se todos os procedimentos que fornecem funcionalidades para a interface gráfica deste módulo.
- **PADS\_DM:** Gerado automaticamente por uma aplicação implementada no PSI, fornece todos os métodos necessários para realizar consultas e manipulações de dados sobre o modelo relacional deste módulo.
- **PADS\_BR:** Implementa procedimentos relacionados com as regras do negócio.

#### 4.4.4 Interface

Dado que não foram feitas alterações ao modelo de dados, foi decidido aproveitar as interfaces já elaboradas. As seguintes figuras apresentam as interfaces usadas pelo processo.

- **Criação de um PAD** - A figura [4.15](#) representa a primeira fase de criação de um PAD. Neste formulário é seleccionada a moeda e tipo de despesa. Na segunda fase,



## Implementação

figura 4.16 são seleccionados diversos tipos de parâmetros, alguns obrigatórios, outros variando consoante o tipo de despesa.

**Pedido de Autorização de Despesa - Criação**

Data: 2009-06-22  
Autor: Susana Pinho

Tipo de Despesa: Normal  
Moeda: Euro

[Criar PAD](#)

[Versão para Impressão](#) [Recomendar Página](#) [Adicionar aos Favoritos](#) [Partilhar](#)

Copyright 1996-2009 © Faculdade de Engenharia da Universidade do Porto [Termos e Condições](#) [Acessibilidade](#) [Índice A-Z](#)  
Página gerada em 2009-06-22 às 17:56:59

Figura 4.15: Criação do pedido de autorização de despesa fase 1

**Pedido de Autorização de Despesa - Criação**

Os campos a **Vermelho** são de preenchimento obrigatório.

N° do PAD: 1697  
Tipo de Despesa: Normal  
Moeda: Euro  
Compra ao Estrangeiro: ☐

Descrição da Despesa: Exemplo de um pedido de autorização de despesa

Responsável: 248271 António Cunha

Centro de Custos: VCIC02 Prestação de Serviços - IRICUP - Expansão do SIFEUP

Fornecedor:

NOTA: se não encontrar o fornecedor, poderá registar mais tarde um Pedido de Criação de Fornecedor.

Observações:

Descrição	IVA	Quant.	Valor Unit. S/IVA	Valor Unit. C/IVA	Valor Total Est.	Documentos
Exemplo	0%	1	12,00 EUR	12,00 EUR	12,00 EUR	<a href="#">Adicionar</a>
<b>Valor Total:</b>			12,00 EUR	12,00 EUR		

NOTA: nas compras ao estrangeiro os valores devem ser inseridos sem IVA

[Submeter](#) [Bem a Reparação](#)

Figura 4.16: Criação do pedido de autorização de despesa fase 2

- **Visualização de um PAD** - Este formulário 4.17 apresenta os detalhes de um PAD. Na primeira parte é apresentado as informações relevantes do PAD e produtos requisitados. Em seguida as actividades que correspondem aos estados por onde o processo atravessou e estado actual apresentando o resultado.
- **Edição de detalhes contabilísticos** - Este formulário apresenta a edição dos dados contabilísticos.
- **Visualização de detalhes contabilísticos** - A figura 4.19 apresenta a visualização dos detalhes contabilísticos.
- **Inserção de facturas** - Nesta formulário 4.20 é permitido inserir as facturas.
- **Gestor de tarefas** - Permitem aceder e responder a todas as tarefas e notificações que são enviadas por instâncias BPEL, à semelhança do que é efectuado pelo Oracle Worklist (figura 4.21).

## Pedido de Autorização de Despesa

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	Susana Pinho
Responsável:	António Cunha
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRICUP - Expansão do SiFEUP à U.P.

Produtos / Serviços Requisitados						
Descrição	IVA	Quant.	Valor Unit. S/IVA	Valor Unit. C/IVA	Valor Total Est.	
Exemplo	0%	1	12,00 EUR	12,00 EUR	12,00 EUR	
Valor Total:					12,00 EUR	

## Actividades do PAD

Estado	Início	Fim	Resultado	Observações
Validação do responsável da despesa do PAD	2009-06-22 18:01:33	em curso...		
início	2009-06-22 18:01:33	2009-06-22 18:01:33	-	

Figura 4.17: Visualização dos detalhes de pedido de autorização de despesa

## Gestão de PADs

### Pedido de Autorização de Despesa - Tratamento Contabilístico

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	Susana Pinho
Responsável:	António Cunha
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRICUP - Expansão do SiFEUP à U.P.

Produtos / Serviços Requisitados						
Descrição	IVA	Quant.	Valor Unit. S/IVA	Valor Unit. C/IVA	Valor Total Est.	
Exemplo	0%	1	12,00 EUR	12,00 EUR	12,00 EUR	
Valor Total:					12,00 EUR	

## Contabilidade

Origem:	
Procedimento de Adjudicação:	
Base Legal:	DL 197/99
Número do Processo:	
Número Geral do Processo:	
Diário CPAUD:	
N. Transacção CPAUD:	
Diário:	
Nº Registo:	

Classificação Económica				
POCED	RCE	F. Financ.	Valor (€)	Observações
Adicionar Rubrica				

Terceiros					
T.Registo	Grupo	Entidade	T.Doc.	Valor (€)	Nº.Registo
Adicionar					

Retenções, Garantias ou Cauções				
Grupo	Entidade	Tipo de Doc.	Valor (€)	Observações
Adicionar RGC				

Guardar Dados

Figura 4.18: Edição de detalhes contabilísticos

## Implementação

### Pedido de Autorização de Despesa

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	Susana Pinho
Responsável:	António Cunha
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRIUP - Expansão do SIFEUP à U.P.

Produtos / Serviços Requisitados				
Descrição	IVA	Valor Total Est.	Valor Total S/IVA	Valor Total C/IVA
Exemplo	0%	12,00 EUR	12,00 EUR	12,00 EUR
Valor Total:		12,00 EUR	12,00 EUR	12,00 EUR

### Contabilidade

Procedimento de Adjudicação:	Conc Limitado p/ Qualif (art 80 nº 1)
Número do Processo:	123456
Número Geral do Processo:	123456
N.Transacção CPAUD:	123456
Diário:	12
Nº Registo:	12

Figura 4.19: Visualização dos detalhes contabilísticos

### Gestão de PADs

#### Pedido de Autorização de Despesa - Inserção de Facturas

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	Susana Pinho
Responsável:	António Cunha
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRIUP - Expansão do SIFEUP à U.P.

Produtos / Serviços Requisitados				
Descrição	IVA	Valor Total Est.	Valor Total S/IVA	Valor Total C/IVA
Exemplo	0%	12,00 EUR	12,00 EUR	12,00 EUR
Valor Total:		12,00 EUR	12,00 EUR	12,00 EUR



Para que o processo possa prosseguir, deve enviar o original dos documentos para os SEF.

Submeter

FEUP - Inserção de um documento - Mozilla Firefox

http://oradev.fe.up.pt:7777/feup\_si/pads\_docs.doc\_editpct\_pad\_it

### Inserção de um documento

Número:

Data:

Tipo: Factura

Ficheiro:  Procurar...

Observações:

Inserir Documento

Concluído

Figura 4.20: Inserção de facturas

### Tarefas Utilizador

#### Filtros

Tipo de Processo:

Estado:

#### Lista de Notificações

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_Compromisso	Pode efectuar a compra Pad 1697	22-06-2009 18:10		3	Efectuada Cancelada

Actualizar

Figura 4.21: Exemplo de notificação no gestor de tarefas

## Capítulo 5

# Resultados

Este capítulo apresenta a última fase de avaliação do projecto, realizado após a implementação dos requisitos. Apresenta e demonstra a satisfação dos resultados obtidos.

Será apresentado um exemplo de utilização, ilustrando o seu funcionamento geral e com especial importância para a sincronização entre o módulo desenvolvido e a tecnologia BPEL.

Na parte final, será feita uma comparação entre o processo em BPEL e Oracle Workflow, sendo evidenciado as vantagens que a passagem BPEL trouxe, bem como, as desvantagens inerentes.

### 5.1 Avaliação Global do Projecto

O trabalho desenvolvido encontra-se em estado avançado, faltando pouco para o seu término. O modelo de dados está concluído e o processo em BPEL está praticamente pronto. Neste último, falta alguns tratamentos de erros e protecção contra situações anormais.

A parte deste projecto que se encontra mais incompleta corresponde à camada de apresentação do módulo de lógica. Desta forma, o que falta elaborar consiste essencialmente nas páginas auxiliares, dentro das quais se destacam a pesquisa e listagem dos processos de pedido de autorização de despesa.

Com a revisão e actualização da camada de interligação, obteve-se uma maior flexibilidade e integração com o motor do BPEL e o sistema de informação. Esta modernização representa um progresso na utilização das mais recentes tecnologias aliadas à arquitectura distribuída e seguindo os conceitos de SOA.

## 5.2 Caso de Utilização

Nesta secção é apresentado um exemplo de utilização do módulo pedidos de autorização de despesa. Para tal, um utilizador criou um novo PAD que posteriormente irá percorrer todo o fluxo, sendo apresentado todas as interações com os utilizadores até ao fim do processo. Este exemplo de utilização é composto pelas seguintes etapas:

1. Um utilizador após identificar os itens que necessita adquirir cria um pedido de autorização de despesa.
  - (a) Selecciona o tipo de despesa, normal e a moeda, EURO, em que se realizará, pressionando em criar Pad para passar à fase seguinte (figura 5.1).

Figura 5.1: Criação do pedido de autorização de despesa parte 1

- (b) Insere os campos obrigatórios: descrição da despesa, responsável, centro de custos e os produtos requisitados. Para o processo continuar, pressiona-se em submeter (figura 5.2).

Figura 5.2: Criação do pedido de autorização de despesa parte 2

## Resultados

3. É apresentado os detalhes do pedido. Inicia-se o processo, pressionando em enviar PAD BPEL (figura 5.3)


### Pedido de Autorização de Despesa

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	<a href="#">Susana Pinho</a>
Responsável:	<a href="#">António Cunha</a>
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRICUP - Expansão do SiFEUP à U.P.

Produtos / Serviços Requisitados						
Descrição	IVA	Quant.	Valor Unit.	S/IVA	Valor Unit. C/IVA	Valor Total Est.
Exemplo	0%	1	12,00 EUR		12,00 EUR	12,00 EUR
Valor Total:						12,00 EUR



O Fornecedor deste PAD não está definido! O botão Editar Dados permite-lhe definir o Fornecedor. Se o Fornecedor que pretende utilizar não existe, por favor registe um [Pedido de Criação de Fornecedor](#).

Figura 5.3: Detalhes do pedido de autorização de despesa

3. É criada uma nova instância BPEL do processo. É apresentado ao utilizador a página do PAD onde se observa o estado do processo - actividade do PAD na figura 5.4.

### Pedido de Autorização de Despesa

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	<a href="#">Susana Pinho</a>
Responsável:	<a href="#">António Cunha</a>
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRICUP - Expansão do SiFEUP à U.P.

Produtos / Serviços Requisitados						
Descrição	IVA	Quant.	Valor Unit.	S/IVA	Valor Unit. C/IVA	Valor Total Est.
Exemplo	0%	1	12,00 EUR		12,00 EUR	12,00 EUR
Valor Total:						12,00 EUR

### Actividades do PAD


Estado	Início	Fim	Resultado	Observações
Validação do responsável da despesa do PAD 	2009-06-22 18:01:33	em curso...		
início	2009-06-22 18:01:33	2009-06-22 18:01:33	-	

Figura 5.4: Detalhes do pedido de autorização de despesa

4. Dado que o autor não corresponde ao responsável, é enviada uma notificação ao responsável da despesa que decide aprová-lo. Deste modo, a tarefa é actualizada e removida automaticamente do Gestor de Tarefas (figura 5.5).

## Resultados

### Tarefas Utilizador

#### Filtros

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoResponsavel  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

#### Lista de Notificações

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_AutorizacaoResponsavelDespesa	Validação de Dados do PAD 1697	22-06-2009 18:1		3	<span>Rejeitar</span> <span>Aprovar</span>

Actualizar

Figura 5.5: Notificação de validação do responsável da despesa

5. Em seguida, uma notificação é enviada para o responsável do centro de custos que aprova a despesa (figura 5.6).

### Tarefas Utilizador

#### Filtros

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoResponsavel  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

#### Lista de Notificações

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_ValidacaoCentroCusto	Autorização da Despesa pelo Centro de Custos Pad 1697	22-06-2009 18:3		3	<span>Rejeitar</span> <span>Aprovar</span>

Actualizar

Figura 5.6: Notificação de validação do responsável da centro de custo

6. A aprovação pelo responsável do centro de custo permite que os utilizadores da contabilidade recebam uma notificação para que se seja efectuado o cabimento(figura 5.7).

### Tarefas Utilizador

#### Filtros

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoResponsavel  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

#### Lista de Notificações

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_CabimentoCompromisso	Efectuar cabimento/compromisso Pad 1697	22-06-2009 18:4		3	<span>Efectuado</span> <span>Rejeitado</span>

Actualizar

Figura 5.7: Notificação de contabilidade informando o que é necessário efectuar o processo de cabimentação

- (a) Acede-se à página do pad e clica-se em editar detalhes contabilísticos.
- (b) É inserido os valores do cabimento e guardado (figura 5.8).

## Resultados

### Gestão de PADs

#### Pedido de Autorização de Despesa - Tratamento Contabilístico

Nº do PAD:	1697
Descrição:	Exemplo de um pedido de autorização de despesa
Tipo de PAD:	Normal
Autor:	Susana Pinho
Responsável:	António Cunha
Centro de Custos:	531CIC - Prestações de Serviços - CICA
Subcentro de Custos:	VCIC02 - Prestação de Serviços - IRIJUP - Expansão do SIFEUP à U.P.

Produtos / Serviços Requisitados						
Descrição	IVA	Quant.	Valor Unit. S/IVA	Valor Unit. C/IVA	Valor Total Est.	
Exemplo	0%	1	12,00 EUR	12,00 EUR	12,00 EUR	
Valor Total:					12,00 EUR	

### Contabilidade

Origem:	
Procedimento de Adjudicação:	
Base Legal:	DL 197/99
Número do Processo:	
Número Geral do Processo:	
Diário CPAUD:	
N.Transacção CPAUD:	
Diário:	
Nº Registo:	

Classificação Económica				
POC-ED	RCE	F. Financ.	Valor (€)	Observações
Adicionar Rubrica				

Terceiros						
T.Registo	Grupo	Entidade	T.Doc.	Valor (€)	Nº.Registo	Nº.Inter
Adicionar						

Retenções, Garantias ou Cauções				
Grupo	Entidade	Tipo de Doc.	Valor (€)	Observações
Adicionar RGC				

Guardar Dados

Figura 5.8: Cabimentação da despesa

(c) No gestor de tarefas, clica-se em efectuado, fazendo prosseguir o fluxo (figura 5.7).

7. Após todas estas validações, são notificados os utilizadores do conselho administrativo que aprovam a despesa (figura 5.9).

### Tarefas Utilizador

#### Filtros

Tipo de Processo:	Todos
	PADS_AutorizacaoConselho
	PADS_AutorizacaoRespons
	PADS_RegistoTerceiros
Estado:	Por Responder
	Respondidas

#### Lista de Notificações

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_AutorizacaoConselhoAdministrativo	Autorização da Despesa pela Instituição	22-06-2009 18:9		3	Rejeitar Aprovar

Actualizar

Figura 5.9: Notificação de validação do conselho administrativo



## Resultados

8. O autor recebe a notificação a informar que pode efectuar a compra. Desta forma, efectua a despesa, e actualiza no gestor de tarefa (figura 5.10).

### Tarefas Utilizador

**Filtros**

Tipo de Processo:   
PADS\_Compromisso  
PADS\_EditPad

Estado:   
Respondidas

**Lista de Notificações**

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_Compromisso	Pode efectuar a compra Pad 1697	22-06-2009 18:10		3	<input type="text" value="Efectuada"/> Cancelada

Actualizar

Figura 5.10: Notificação de autor para efectuar a compra

9. Após a compra, o autor efectua a inserção de facturas online e manda seguir o processo no seu fluxo. Envia as facturas em papel para o serviços económico-financeiros (figura 5.11).

## Gestão de PADs

### Pedido de Autorização de Despesa - Inserção de Facturas

Nº do PAD: 1697

Descrição: Exemplo de um pedido de autorização de despesa

Tipo de PAD: Normal



Autor: [Susana Pinho](#)


Responsável: [António Cunha](#)

Centro de Custos: 531CIC - Prestações de Serviços - CICA

Subcentro de Custos: VCIC02 - Prestação de Serviços - IRICUP - Expansão do SiFEUP à U.P.

☐

Produtos / Serviços Requisitados					
Descrição	IVA	Valor Total Est.	Valor Total S/IVA	Valor Total C/IVA	Documentos
Exemplo	0%	12,00 EUR	<input type="text" value=""/>	<input type="text" value=""/>	 
Valor Total:		12,00 EUR			

 Para que o processo possa prosseguir, deve enviar o original dos documentos para os SEF.

Submeter

Figura 5.11: Inserção online de facturas

10. Os serviços económico-financeiros recebem a notificação do processo de inserção de facturas, validam as facturas em papel e mandam prosseguir o processo (figura 5.12).
11. É enviado uma notificação à contabilidade informando que é necessário efectuar o registo de terceiros (figura 5.13).

## Resultados

### Tarefas Utilizador

**Filtros**

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoRespons  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

**Lista de Notificações**

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_RegistoTerceiros	Espera chegada das facturas em papel Pad 1697	25-06-2009 12:59		3	<span>Terminado</span>

Actualizar

Figura 5.12: Notificação de registo de terceiros

### Tarefas Utilizador

**Filtros**

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoRespons  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

**Lista de Notificações**

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_RegistoTerceiros	Efectuar Obrigação/Registo de Terceiros Pad 1697	22-06-2009 18:14		3	<span>Registado</span>

Actualizar

Figura 5.13: Notificação de registo de terceiros

- (a) Accede-se à página do pad e clica-se em editar detalhes contabilísticos.
  - (b) Na secção terceiros, é inserido o registo e em seguida guardado.
  - (c) No gestor de tarefas, clica-se em efectuado, fazendo prosseguir o fluxo.
12. Em seguida, é enviada a notificação aos utilizadores da tesouraria para fazerem a integração da ordem de pagamento. Depois de integrada, actualiza-se o estado no gestor tarefas (figura 5.14).

**Tarefas Utilizador**

**Filtros**

Tipo de Processo: Todos  
PADS\_AutorizacaoConselho  
PADS\_AutorizacaoRespons  
PADS\_RegistoTerceiros

Estado: Por Responder  
Respondidas

**Lista de Notificações**

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_Pagamento	Integração da Ordem de Pagamento Pad 1697	22-06-2009 18:15		3	<span>Efectuada</span>

Actualizar

Figura 5.14: Integração da ordem de pagamento

13. Por fim, há a notificação aos utilizadores da tesouraria que aprovam a ordem de pagamento (figura 5.15).

## Resultados

**Tarefas Utilizador**

**Filtros**

Tipo de Processo:

Estado:

**Lista de Notificações**

Tipo	Título	Data Criação	Expira em	Prioridade	Opções
PADS_Pagamento	Autorização da Ordem de Pagamento Pad 1697	22-06-2009 18:16		3	<input type="text" value="Nao Autorizado"/> <input type="text" value="Autorizado"/>

Figura 5.15: Autorização do pagamento

### 5.3 Comparação entre BPEL e Workflow

O Workflow está integrado com SIGARRA o que leva a que todas as informações de um dado processo sejam mantidas do lado do workflow. Esta abordagem permite a não replicação de dados. Contudo, o workflow necessita de guardar um conjunto de informações não relevantes para o utilizador. Dado que as informações relevantes de um processo não podem ser desassociadas das informações não relevantes e consequentemente não eliminadas, leva a um aumento progressivo dos volumes de dados tornando o acesso a estes cada vez mais lento ao longo do tempo. Para não se cometer esta má decisão mais uma vez, foi decidido na implementação deste projecto, fazer uma separação da informação. Todos os dados relevantes, no decorrer do processo, são guardados no SIGARRA sem nenhuma dependência com a base de dados BPEL. Desta forma, após a conclusão de um processo, a instância BPEL pode ser eliminada, visto que toda a informação relevante se encontra guardada no SIGARRA.

Ao nível da arquitectura houve a separação do processo em diversos subprocessos, o que permite uma maior facilidade no desenvolvimento ao nível de organização do código, bem como manutenção de todo o processo. Futuras alterações tornam-se desta forma mais simples. Mas a vantagem mais importante é a possibilidade de aproveitamento dos subprocessos por parte de outros futuros processos.

Um vantagem inerente ao BPEL será que é um standard aberto enquanto que o Workflow é um aplicação proprietária. Futuras mudanças ao nível do servidor possibilitam que a conversão do processo seja simples, fácil e rápida, comparativamente com Workflow que necessitaria de um reconstrução total.

O Workflow é desenvolvido em PL/SQL, a mesma linguagem que o SIGARRA. Desta forma, integração entre ambos os sistemas é simples, rápida e directa. Comparativamente, para ocorrer comunicação entre o SIGARRA e o BPEL é necessário uma camada extra, comunicação via SOAP, o que implica recursos adicionais. Apesar desta aparente desvantagem, permite que o BPEL seja capaz de interagir facilmente com sistemas heterogéneos.

Com a passagem do processo de pedidos de autorização de despesa de Workflow para BPEL foi alcançado mais um passo na direcção de tornar o sistema de informação mais distribuído baseado nos conceitos de arquitectura orientada a serviços.

## **Capítulo 6**

# **Conclusões e Perspectivas de Trabalho Futuro**

Este capítulo apresenta uma apreciação final e global de satisfação dos objectivos propostos para este projecto, e as respectivas direcções para uma futura evolução das soluções e funcionalidades implementadas.

### **6.1 Satisfação dos Objectivos**

A implementação deste projecto representa mais um passo na modernização do sistema de informação da Faculdade de Engenharia da Universidade do Porto que, aliado a todas as vantagens de uma arquitectura distribuída e, seguindo os conceitos de arquitectura orientado a serviços, permite revolucionar as aplicações envolvidas como foi o caso do módulo de pedidos de autorização de despesa.

Se fizermos uma avaliação global ao projecto desenvolvido, observa-se que os requisitos fundamentais foram cumpridos de forma adequada. Foi revista e completada a infraestrutura de comunicação entre o Oracle BPEL e o SIGARRA. Devido à longa duração desta tarefa, não foi concluído na totalidade a conversão deste processo. Mesmo assim, as funcionalidades ainda por implementar, não interferem com o funcionamento básico do processo.

Por fim, a conversão do módulo de Pedidos de Autorização de Despesa de workflow para BPEL, o processo mais complexo, demonstra a capacidade desta nova tecnologia em lidar com os mais complicados e delicados processos.

## **6.2 Trabalho Futuro**

Em termos de desenvolvimentos futuros, este deverá passar pela finalização da camada de apresentação do módulo de lógica. Este constituiu um conjunto de páginas web de acessos complementares ao sistema dos quais se destaca a pesquisa de PADs.

Outra situação a resolver será os poucos tratamentos de erros e excepções anormais que ainda faltam resolver.

Finalmente, um dos desenvolvimentos interessantes será a visualização do estado de um pedido de autorização de despesa por parte de um fornecedor. Isto permitirá que um fornecedor esteja ocorrente do estado do seu processo, sem ser preciso recorrer aos métodos tradicionais: contacto telefónico ou por email.

# Referências

- [dEdUdP09] Faculdade de Engenharia da Universidade do Porto. Serviços económico-financeiros, 2009. [http://www.fe.up.pt/si/unidades\\_geral.visualizar?p\\_unidade=14](http://www.fe.up.pt/si/unidades_geral.visualizar?p_unidade=14).
- [dR] Assembleia da República. Lei n.º 108/88 de 24 de setembro - autonomia das universidades. <http://intranet.uminho.pt/Arquivo/Legislacao/AutonomiaUniversidades/L108-88.pdf>.
- [dRM08] Ivo André da Rocha Marinho. Utilização do oracle bpm process manager, 2008. Faculdade de Engenharia da Universidade do Porto.
- [Jos07] Nicloai M. Josuttis. *SOA in Practice: The Art of Distributed System Design*. O'Reilly, First edition, 2007.
- [Jur06] Matjaz B. Juric. *Business Process Execution Language for Web Services*. Packt Publishing, Second edition, 2006.
- [Ken07] Mark Kennedy. Oracle bpm process manager administrator guide 10g, January 2007. [http://download-west.oracle.com/docs/cd/B31017\\_01/integrate.1013/b28982.pdf](http://download-west.oracle.com/docs/cd/B31017_01/integrate.1013/b28982.pdf).
- [McL08] Michael McLaughlin. *Oracle Database 11g PL/SQL Programming*. The McGraw-Hill Companies, Inc., 2008.
- [San06] Dra. Lúcia Santos. Acção de formação - poc educação, 2006.

## Anexo A

# Diagramas

### A.0.1 Diagrama de fluxo

O diagrama [A.1](#) apresentado na figura foi elaborado por forma a especificar e representar o fluxo de actividades que o novo processo de negócio do módulo de Pedidos de Autorização de Despesa possui.

### A.0.2 Modelo de Dados

Relativamente ao modelo relacional de dados definido e utilizado pelo módulo de Pedidos de Autorização de Despesa, é apresentado na figura [A.2](#) todo o esquema de tabelas, registos e dependências existentes. É igualmente possível identificar na figura [A.3](#) a inclusão de registos do estado processo usados, tais como o identificador da instância, que permitem a interligação do SIGARRA com a tecnologia BPEL.

## Diagramas

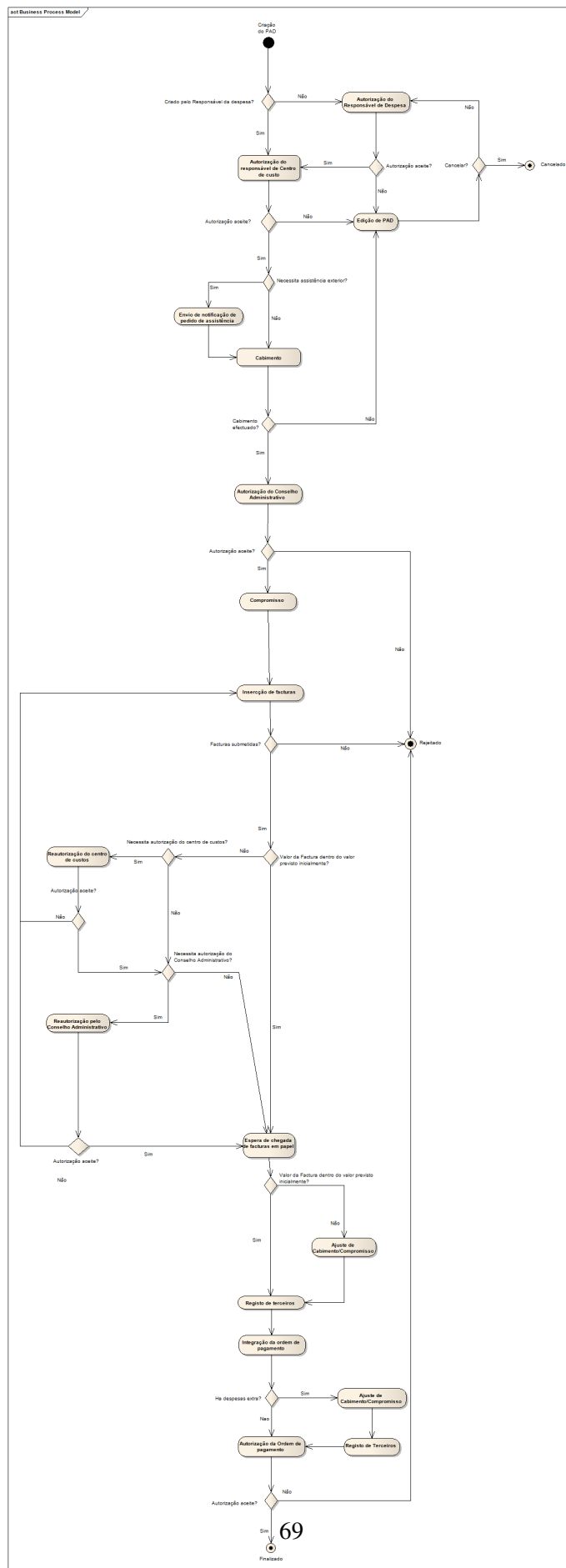


Figura A.1: Diagrama de fluxo do processo de Pedidos de Autorização de despesa



## Diagramas

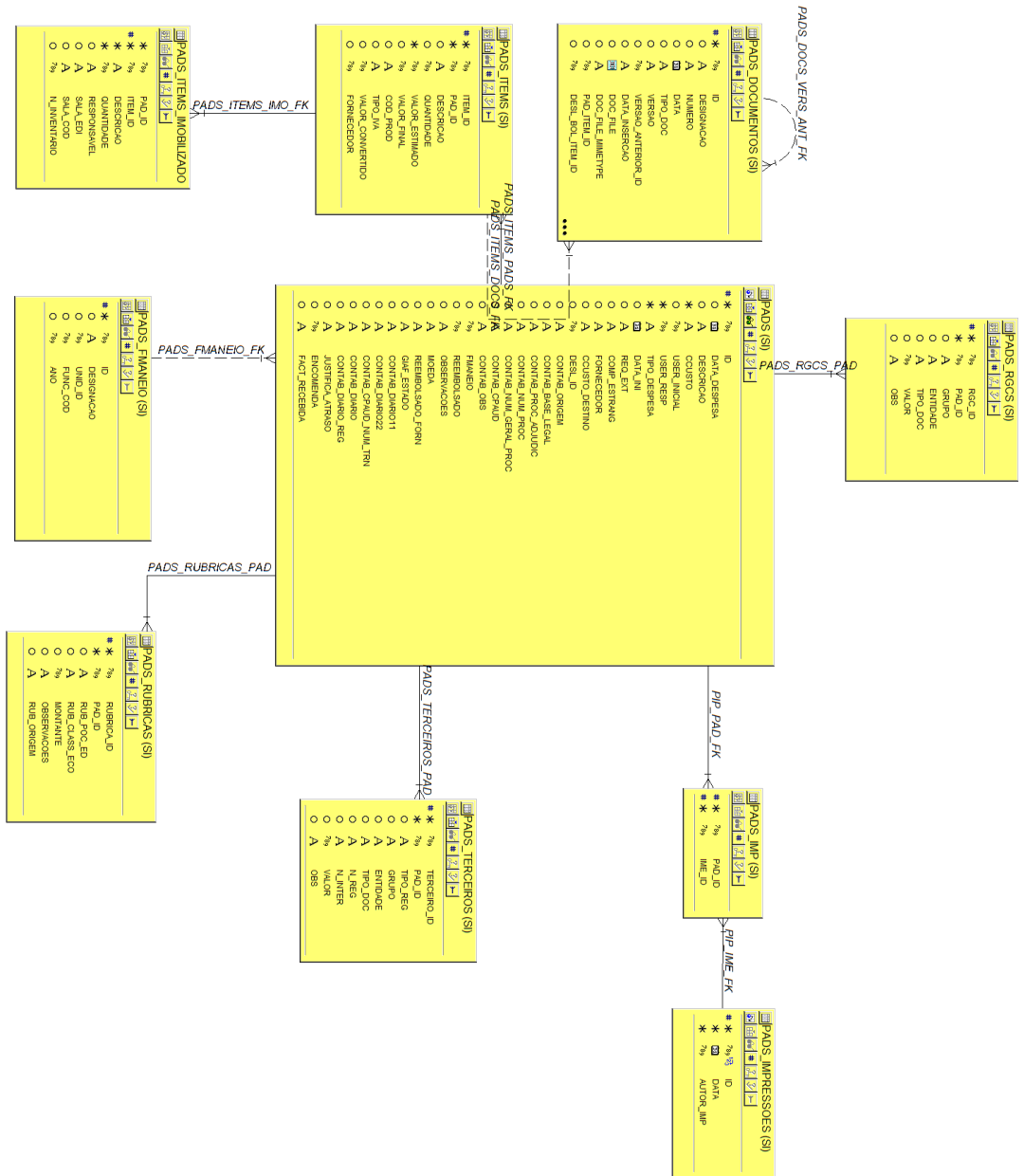


Figura A.2: Modelo de dados do módulo dos Pedidos de Autorização de Despesa

## Diagramas

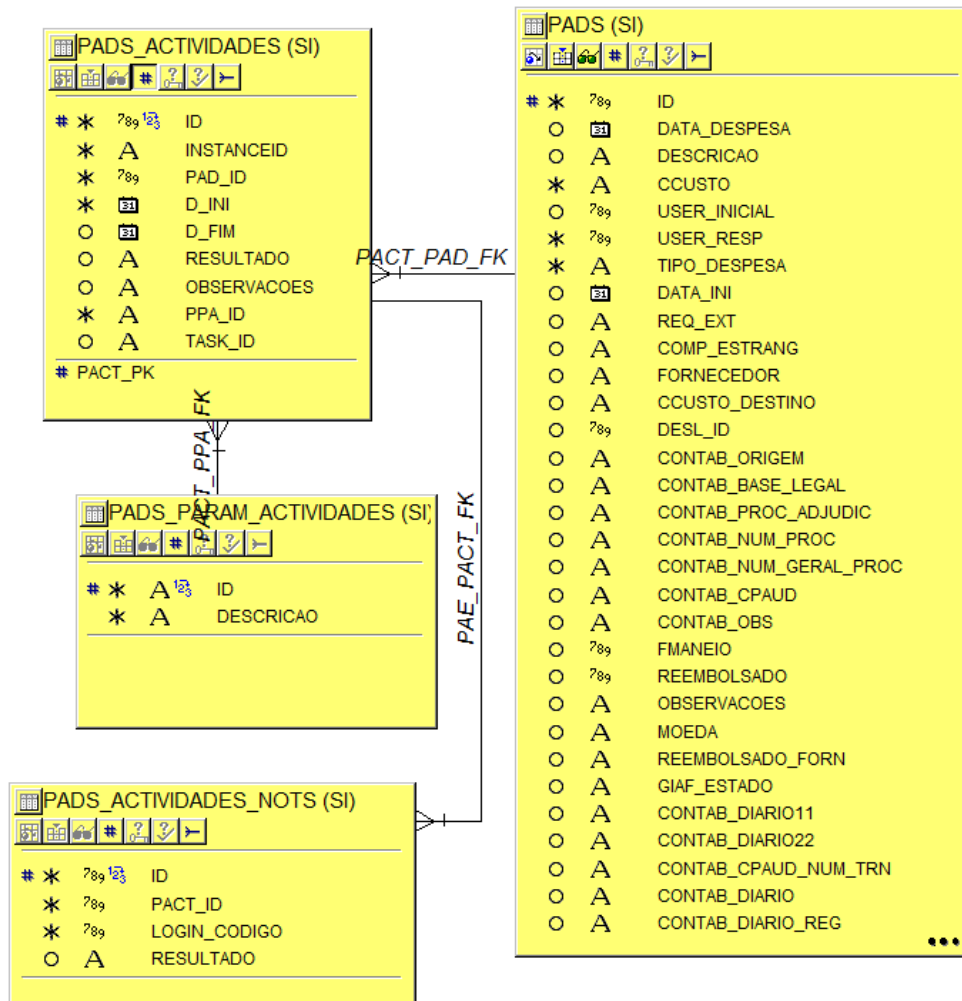


Figura A.3: Modelo de dados da integração do bpeI com o módulo dos Pedidos de Autorização de Despesa